PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-090076
(43)Date of publication of application : 31.03.2000

(51)Int.Cl. G06F 17/21
G06F 12/00
G06F 17/30

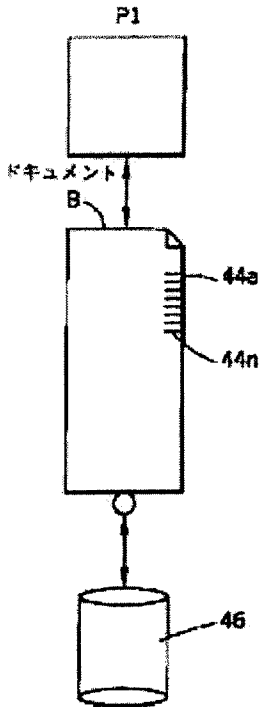
(21)Application number : 11-244025
(22)Date of filing : 30.08.1999

(71)Applicant : XEROX CORP
(72)Inventor : PETERSEN KARIN
DOURISH JAMES P
EDWARDS WARREN K
LAMARCA ANTHONY G
LAMPING JOHN O
SALISBURY MICHAEL P
TERRY DOUGLAS B
THORNTON JAMES D

(30)Priority
Priority number : 98 143551 Priority date : 31.08.1998 Priority country : US

(54) METHOD AND SYSTEM FOR MANAGING DOCUMENT

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a method and a system for managing user level document based on property.
SOLUTION: A user level managing mechanism is inserted to the read/write route of a computer system. The mechanism is packaged as a property 44 added to a document B. The document B having the added property 44 has a function for separating contents 46 of the document B from the property 44 for describing the document B. Thus, since the contents of the document are separated from the document property, the access and management of the user level to the property is realized and when programming, storing and retrieving the document, the user can have flexibility as a result. Thus, the user can constitute a document group in which one document can appear in plural groups. The property is peculiar to the user and document in the meaning of relating the management of the specified document to the intended user through the addition.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（A）**　(11)特許出願公開番号
特開2000－90076
（P2000－90076A）
(43)公開日　平成12年３月31日(2000.3.31)

(51)Int.Cl.⁷	識別記号	FI		テーマコード（参考）
G06F 17/21		G06F 15/20	590E	
12/00	520		520E	
17/30		15/20	570N	
			570D	
			596A	

審査請求　未請求　請求項の数３　ＯＬ　（全 22 頁）　最終頁に続く

(21)出願番号　特願平11－244025

(22)出願日　平成11年８月30日(1999.8.30)

(31)優先権主張番号　０９／１４３５５１

(32)優先日　平成10年８月31日(1998.8.31)

(33)優先権主張国　米国（ＵＳ）

(71)出願人　590000798
ゼロックス　コーポレイション
ＸＥＲＯＸ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国　06904－1600　コネティカット州・スタンフォード・ロング　リッチ　ロード・800

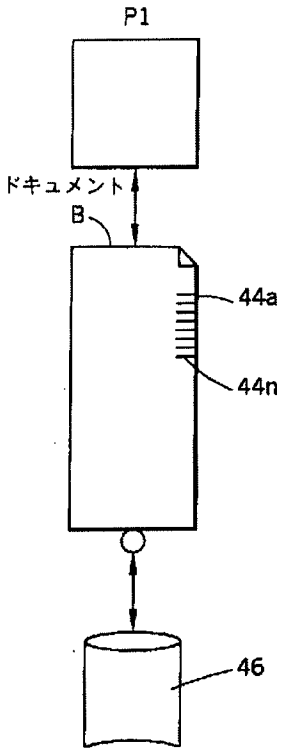
(72)発明者　カリン　ピーターソン
アメリカ合衆国　カリフォルニア州　パロ　アルト　アルマ　ストリート　1335

(74)代理人　100075258
弁理士　吉田　研二　（外２名）

最終頁に続く

(54)【発明の名称】　ドキュメント管理方法およびドキュメント管理システム

(57)【要約】
【課題】　プロパティに基づくユーザレベルドキュメント管理方法およびシステムを提供する。
【解決手段】　ユーザレベル管理機構は、コンピュータシステムの読み取り／書き込み経路に挿入される。機構は、ドキュメント（Ｂ）に付加されるプロパティ４４として実装される。付加されたプロパティ４４を有するドキュメントＢは、ドキュメントＢの内容４６をドキュメントＢを記述するプロパティ４４から分離する機能を有する。このように、ドキュメント内容をドキュメントプロパティから分離することによって、プロパティに対するユーザレベルのアクセスおよび管理が実現され、その結果、ドキュメントの編成、記憶および検索においてユーザが柔軟性を持つことが可能になる。本機構により、ユーザは、一のドキュメントが複数の群において出現可能なドキュメント群を構成することが可能となる。プロパティは、付加されることにより特定のドキュメントの管理を意図するユーザに関連するという意味において、ユーザおよびドキュメントに特有である。



1

【特許請求の範囲】

【請求項１】　命令を発行する少なくとも一つのアプリケーションと、ドキュメントを記憶する少なくとも一つのデータ記憶保管場所とを備えるコンピュータシステムのドキュメント管理システムを用いて、ドキュメントを管理する方法であって、

前記コンピュータシステムの第一ユーザの前記ドキュメント管理システムのプロパティへのアクセスを可能にするステップと、

前記第一ユーザの操作に基づいて、プロパティのうち第一に選択されるプロパティを、ドキュメント管理システムの第一のドキュメントに付加するステップと、

前記付加された第一選択プロパティを記憶するステップと、

前記第一ドキュメントの内容を、前記第一選択プロパティが記憶されるロケーションと分離して記憶するステップと、

前記ドキュメントの内容をドキュメントのプロパティと分離して管理するステップと、

前記付加された第一選択プロパティのうち少なくとも一つを用いて、第一ドキュメントを検索するステップと、を含み、前記検索するステップは、第一ドキュメントの内容を検索するステップを含むことを特徴とするドキュメント管理方法。

【請求項２】　請求項１に記載の方法において、さらに、

第二ユーザの前記プロパティへのアクセスを可能にするステップと、

第二ユーザにの操作に基づいてプロパティのうち第二に選択されるプロパティを第二ドキュメントに付加するステップであって、少なくとも一つの前記第二選択プロパティは前記第一選択プロパティと異なり、前記第二ドキュメントの内容は前記第一ドキュメント内容であるステップと、

前記付加された第二選択プロパティを記憶し、それによって、前記第一ドキュメントの内容である第二ドキュメントの内容が、前記第二ドキュメントのプロパティと分離して記憶するステップと、

前記第二選択プロパティを、前記第一選択プロパティとは独立に管理するステップと、を含むことを特徴とするドキュメント管理方法。

【請求項３】　ドキュメント管理システムであって、

複数のユーザが前記システムを使用できるように構成されるシステムユーザインターフェースと、

複数のプロパティを含むドキュメント管理層と、

前記プロパティの中から選択されるプロパティを、選択されるドキュメントに付加する機構であって、前記システムのユーザによって管理されるプロパティ付加機構と、

前記ドキュメントに付加されるプロパティと前記ドキュメントの内容とを別々のロケーションに記憶する機構と、

前記付加プロパティに基づいて前記ドキュメントを検索する機構と、を含むことを特徴とするドキュメント管理システム。

2

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、ドキュメント管理システムの技術に関し、特に、システムの読み取り／書き込み経路に介在するユーザレベルで管理される機構によってドキュメント（文書）が編成されかつ管理される分散型ドキュメントインフラストラクチャに関する。

【０００２】

【従来の技術および発明が解決しようとする課題】発明者らは、ユーザが、コンピュータとの大量の対話によって、たとえば、種々の電子ドキュメントの保存、ファイリング、編成および検索など、ドキュメント管理を処理しなければならないことを認識している。これらのドキュメントは、ローカルディスク、ネットワークシステムのファイルサーバ、電子メールのファイルサーバ、ワールドワイドウェブ、または他の様々な場所に存在する。現代の通信配布システムは、ユーザのドキュメント空間内に組み込むことができるドキュメントの流れを大いに増大させる効果を有しているため、蓄積されたドキュメントを見たり対話したりするための一層優れたツールに対する要求が増大している。

【０００３】ドキュメント空間を整理する最も一般的なツールは、階層保管システムとして知られる単一の基本的なメカニズムに依存している。この階層保管システムでは、ドキュメントはディレクトリまたはフォルダ内に存在するファイルとして取り扱われており、これらのディレクトリまたはフォルダ自体は他のディレクトリ内に含まれており、これによりドキュメント空間の対話のための構造を提供する階層が形成されている。ディレクトリ階層内のそれぞれのディレクトリは、一般に多数の個別のファイルを含むことになる。一般的に、ファイルやディレクトリは、ネットワークを経由して共有化される大きな保管容量の中で、英数字の呼び名が与えられる。そのようなネットワークでは、個々のユーザには、特別なディレクトリが割り当てられる。

【０００４】サブディレクトリ内に配置されたファイルは、複合のパスネームが割り当てられる。例えば、文字列D:¥TREE¥LIMB¥BRANCH¥TWIG¥LEAF.FILは、ファイルLEAF.FILの位置を記述することができ、このファイルLEAF.FILの直接のディレクトリはTWIGであり、文字Ｄで識別されるドライブ上の深いファイル階層に位置している。それぞれのディレクトリ自体は、ファイル名、サイズ、位置データ、およびファイル作成または更新の日時を含むファイルである。

【０００５】ファイルシステムにおけるナビゲーション

3

は、大きくとらえると、ファイル階層上にマップされている語義構造（semantic structures）におけるナビゲーションとして考えることができる。そのようなナビゲーションは、通常はブラウザやダイアログボックスを用いて達成される。このため、ユーザがファイルシステム内を動き回ってファイル（LEAF.FIL）を得る場合、この動きは、一つのファイルまたはフォルダから他への動きとしてだけではなく、徐々に小さな位置に集束するドキュメントの特徴を利用した検索手順として、見ることができる。階層はファイル整理に利用可能なほぼ唯一のメカニズムであるため、検索の構造はファイルシステムにより提供される階層上にマップされる。しかしながら、ドキュメントとファイルは同じものではない。

【０００６】ファイルはディレクトリによってグループ化されるので、単一のドキュメントを内容の異なるいくつかのグルーピングで関連付けることは厄介である。ディレクトリ階層は、階層の各ノードに配置されたアクセスコントロールによって、ドキュメントへのアクセスをコントロールするためにも使用される。このことはファイルアクセスを一人または数人にのみ許可することを困難にする。本発明では、種々のドキュメントの集合における帰属関係を含むドキュメント固有の識別をそのプロパティ（特性）から分離することによって、これらの問題を解消する。

【０００７】従来の他の欠点として、従来の階層ファイルシステムが「単一の相続」構造を有する点があげられる。特に、ファイルは一度に一つの場所にのみ存在可能とされるため、語義構造の中では一つの場所のみしか占有できない。リンクおよびエイリアスの使用は、そのような制限を改良するための試行である。従って、ファイルの整理が必要な構造についてのユーザの構想が時間の経過につれて変化する間に、上述した階層は固定され強固になってしまう。個々のファイルをそのような構造内に移動することはかなり簡単な仕事ではあるが、ファイルの大きなセットの再整理は、非常に複雑で、効率が悪く、かつ時間がかかる。上述したことから、従来のシステムは、時間と共に変化するカテゴリーに基づいてファイル構造を変化させるというユーザの要求に指向していないことが分かる。ある時点では、ユーザはプロジェクトに関してドキュメント空間を整理したいと思うことがあり、一方将来のある時期には、このユーザは時間および／またはドキュメント内容に従って構成を生成したいと希望することがあるかもしれない。厳密な階層構造では、シームレスな手法による複数の観点に対するドキュメント管理が実施されないため、ドキュメント検索の効率が悪化してしまう。

【０００８】また、従来のファイルシステムは、ドキュメントの保管および検索に対して単一のモデルしかサポートしていない。このことは、ドキュメントはその作成者がドキュメントに与えた構造または構想に従って検索

4

されることを意味する。他方、作成者ではないユーザは、ドキュメントが保管された方法とは異なる構想すなわちグルーピングに従って、ドキュメントを検索することを希望することがある。

【０００９】さらに、ドキュメント管理は演算力を有する装置上で行われるので、ドキュメントの整理を支援するためにこの演算力を利用できるという利点がある。例えば、ドキュメントにスペルチェッカのプロパティを付けることによって、ドキュメントは、要求しているアプリケーションに戻される内容が正確に書かれるように、ドキュメントの読取り動作が拡張される。

【００１０】

【課題を解決するための手段】本発明は、命令を発行する少なくとも一つのアプリケーションと、ドキュメントを記憶する少なくとも一つのデータ記憶保管場所とを備えるコンピュータシステムのドキュメント管理システムを用いてドキュメントを管理することを目的とする。この方法は、コンピュータシステムの第一ユーザのドキュメント管理システムのプロパティへのアクセスを可能にするステップと、第一ユーザの操作に基づいて、プロパティのうち第一に選択されるプロパティを、ドキュメント管理システムの第一のドキュメントに付加するステップと、付加された第一選択プロパティを記憶するステップと、第一ドキュメントの内容を、第一選択プロパティが記憶されるロケーションと分離して記憶するステップと、ドキュメントの内容をドキュメントのプロパティと分離して管理するステップと、付加された第一選択プロパティのうち少なくとも一つを用いて、第一ドキュメントを検索するステップと、を含み、検索するステップは、第一ドキュメントの内容を検索するステップを含むことを特徴とする。

【００１１】さらに、本発明のドキュメント管理方法は、第二ユーザのプロパティへのアクセスを可能にするステップと、第二ユーザにの操作に基づいてプロパティのうち第二に選択されるプロパティを第二ドキュメントに付加するステップであって、少なくとも一つの第二選択プロパティは第一選択プロパティと異なり、第二ドキュメントの内容は第一ドキュメント内容であるステップと、付加された第二選択プロパティを記憶し、それによって、第一ドキュメントの内容である第二ドキュメントの内容が、第二ドキュメントのプロパティと分離して記憶するステップと、第二選択プロパティを、第一選択プロパティとは独立に管理するステップとを含むことを特徴とする。

【００１２】また、本発明にかかるドキュメント管理システムは、複数のユーザがシステムを使用できるように構成されるシステムユーザインターフェースと、複数のプロパティを含むドキュメント管理層と、プロパティの中から選択されるプロパティを、選択されるドキュメントに付加する機構であって、システムのユーザによって

管理されるプロパティ付加機構と、ドキュメントに付加されるプロパティとドキュメントの内容とを別々のロケーションに記憶する機構と、付加プロパティに基づいてドキュメントを検索する機構と、を含むことを特徴とする。

【００１３】システムの読み取り／書き込み経路に介在してユーザレベルで管理される機構は、ドキュメントに付属するプロパティとして実装することができる。プロパティはそれを付加し特定のドキュメントの管理を意図とするユーザに関連するという意味で、これらのプロパティはユーザおよびドキュメントに特有である。ドキュメントのプロパティによって記述されるこの構造は、ドキュメント内容のロケーション（記憶位置）を、ドキュメントの管理から分離することを考慮する。プロパティの実装により、ドキュメントの記憶および検索が記憶位置に基づいている従来のファイルシステムおよびフォルダ階層に固執する必要がなくなる。本発明により、ドキュメントからなるドキュメント群にアクセスし、それを共用し、管理する方法が簡単になる。それは、人々が、抽象化のレベルを、たとえば、ディスクドライブ、ファイルサーバ、およびディレクトリ名などの低レベルの概念から離れてより高いレベルのより人間指向型の概念に引き上げることによって実現する。ユーザは高レベルのプロパティをドキュメントと関連付け、一方、これらのプロパティを本発明によるドキュメント管理システムに如何にして最善に与えるかという特定の決定は不要になる。

【００１４】

【発明の実施の形態】本発明を一層詳細に説明する前に、説明の中で使用する用語をここにまとめる。以下に定義を示す。

【００１５】動作（Action）：プロパティの挙動部分。

【００１６】動的プロパティ（Active Property）：ドキュメント管理システム内でドキュメントを変更するためあるいは別の変更を行うためのいずれかのために、そのコードが演算力の使用を許されるプロパティ（特性）。

【００１７】任意の（Arbitrary）：ドキュメントに何らかのプロパティを提供する能力。

【００１８】基本ドキュメント（Base Document）：ドキュメントの本質的なビットに対応する。ドキュメントごとに基本ドキュメントは一つのみである。ドキュメントの内容を決定し、ドキュメントのプロパティを含むことができ、またドキュメントに対する全プリンシパル（後述）の見解の一部である。

【００１９】基本プロパティ（Base Properties）：基本ドキュメントに関連する固有のドキュメントプロパティ（特性）。

【００２０】ビットプロバイダ（Bit Provider）：基本ドキュメントの特別なプロパティ、読取り動作や書込み動作を行って、ドキュメントに内容を提供する。ドキュメントの種々のバージョンまたは内容を暗号化したバージョンの読み込み（fetching）など別の動作を行うこともできる。

【００２１】ブラウザ（Browser）：ユーザがドキュメントを配置したり整理することを許可するユーザインターフェース。

【００２２】群（Collection）：他のドキュメントをその内容として含むドキュメントの形式。

【００２３】合成ドキュメント（Combined Document）：群および内容を含むドキュメント。

【００２４】内容（Content）：手紙の中の文字あるいは電子メールのメッセージの本文など、ドキュメント内に含まれる核心情報。

【００２５】内容ドキュメント（Content Document）：内容を有するドキュメント。

【００２６】分散型（Distributed）：異なるシステム（すなわち、ファイルシステム、WWW、電子メールサービスなど）内のドキュメントの保管をユーザには分からない方法でコントロールするシステムの能力。システムは複数の保管場所に配置されたドキュメントを、ドキュメントの内容が保管されている場所についての知識を持っていることをプリンシパルに要求せずに、プリンシパルに提供することを許可する。

【００２７】ＤＭＳ：ドキュメント管理システム。

【００２８】ドキュメント（Document）：これは特定の内容およびこの内容に付加された何らかのプロパティを参照する。参照された内容は、直接参照でも間接参照でも良い。ＤＭＳの最小要素である。４つのタイプのドキュメント、即ち、群、内容ドキュメント、内容なしのドキュメント、および合成ドキュメントがある。

【００２９】ドキュメントハンドル（Document Handle）：全てに共通の表示態様（view）またはあるプリンシパルの表示態様のいずれかの、ドキュメント上の特定の表示態様に対応する。

【００３０】ドキュメントＩＤ（DocumentID）：各基本ドキュメントに対する独特な識別子。参照ドキュメントは、その指示物からドキュメントＩＤを引く継ぐ。ドキュメントの同一性は、このように参照ドキュメントの参照物と基本ドキュメントとの間を接続することによって確立される。論理的に、単一のドキュメントは基本ドキュメントであると共にそれを参照する参照ドキュメントである。

【００３１】カーネル（Kernel）：ドキュメント上のすべての動作を管理する。プリンシパルは一つ以上のカーネルを持つことができる。

【００３２】マルチプリンシパル（Multi-Principal）：各プリンシパルのプロパティが異なる基本ドキュメント上に自分自身のプロパティのセットを持つための複数のプリンシパルについての能力。

【００３３】通知（Notification）：プロパティおよび外部装置が、ＤＭＳ内の他の場所で発生する動作およびイベントを発見することを可能にする。

【００３４】内容なしのドキュメント（No Content Document）：プロパティのみを含むドキュメント。

【００３５】既製のアプリケーション（Off-the-Shelf Applications）：現在存在するオペレーティングシステムが供給するプロトコルおよびドキュメント保管メカニズムを利用する従来のアプリケーション。

【００３６】プリンシパル（Principal）：ドキュメント管理システムの「ユーザ」。ドキュメント管理システムを利用するそれぞれの人または物はプリンシパルである。人のグループもプリンシパルになり得る。ドキュメント上の各プロパティをプリンシパルに関係付けることができるので、プリンシパルが中心となる。これにより異なるプリンシパルが、同じドキュメント上で異なる配景を持つことができる。

【００３７】プロパティ（Property）：内容に付加できる情報または挙動のビット。内容にプロパティを加えても、内容の同一性は変化しない。プロパティはドキュメント上に配置できるタグであり、各プロパティは名前と値（および任意に呼び出し可能な一組の方法）とを有する。

【００３８】プロパティ発生器（Property Generator）：プロパティをドキュメントの内容から引き出すための特別な場合のアプリケーション。

【００３９】参照ドキュメント（Reference document）：ドキュメントの一つのプリンシパルの表示態様に対応する。基本ドキュメントを参照する（参照ドキュメントＡは基本ドキュメントＢを参照する）と共に、一般に別のプロパティも含む。参照ドキュメントにより付加されたプロパティは、その参照のみに属し、別のプリンシパルがこれ等のプロパティを見るためには、はっきりとそれらのプロパティを要求する必要がある。従って、プリンシパルがその参照ドキュメントを通して見る表示態様は、（基本ドキュメントを通る）ドキュメントの内容であり、また（参照および基本ドキュメントの両方の）プロパティのセットである。基本ドキュメントの所有者でさえ、その基本ドキュメントに対して参照ドキュメントを持つこともでき、所有者はこの基本ドキュメントにドキュメントの個人的なプロパティを配置する。この個人的なプロパティはドキュメントの本質的な部分と考えるべきではなく、また他のすべてのプリンシパルの表示態様の中に配置すべきではない。

【００４０】空間（Space）：プリンシパルが所用するドキュメント（基本または参照）のセット。

【００４１】静的プロパティ（Static Property）：ドキュメントに関係する名前－値の対。動的プロパティとは異なり、静的プロパティには挙動がない。ドキュメントについての検索可能なメターデータ（meta-data）情報を提供する。

【００４２】序文

従来の技術の部分で説明したように、ファイルシステムがファイルを管理するために提供する構造は、それによってユーザがドキュメントを整理しまたドキュメントと対話する構造になる。しかしながら、ドキュメントとファイルは同じものではない。本発明は、ドキュメントに関連するプロパティの管理を、ドキュメントの内容の管理から分離するという直接的な目標を有している。このため、ユーザ指定のドキュメントのプロパティは、このドキュメントが保管されている場所ではなく、ドキュメントの消費者すなわちドキュメントのユーザの近くで管理される。ユーザプロパティの管理をドキュメントコントロール自体から分離することにより、ドキュメント管理のコントロールを、閉鎖したファイルシステムという概念からユーザ中心の方法論に移すことができる。

【００４３】図１は、ドキュメントが階層構造によって記載されたそれらの位置に従って整理される階層保管システムと、ドキュメントがそれらのプロパティ（例えば、作成者＝dourish、タイプ＝paper、ステータス＝draft、など）に従って整理される本発明との間の差異を示している。これは、ドキュメントが例えある位置から他の位置に移動したとしても、プロパティを保持しており、またプロパティの割り当ては良好な細分性を持つことができることを意味している。

【００４４】プロパティを本発明のドキュメント管理システム内に統合するためには、プロパティは、動作の結果を変更する能力ならびに別の動作を行う能力の両方を備えて、内容および／またはプロパティのコンピュータシステムの読取り／書込み経路の中に存在させる必要がある。この構想の概要を図２に示す。図２では、ひとたびユーザ（Ｕ）が動作要求（Ｏ）を発行すると、その動作をオペレーティングシステム（ＯＳ）が実行する前に、コールが本発明のドキュメント管理システム（ＤＭＳ）Ａに対してなされる。このためＤＭＳ（Ａ）は、本発明の意図した構想を達成するように機能することができる。これには、それ自身の動作要求（Ｏ′）を通してオペレーティングシステム（ＯＳ）と対話するＤＭＳ（Ａ）が含まれる。ひとたび動作要求（Ｏ′）が完了すると、結果（Ｒ）がＤＭＳ（Ａ）に戻され、今度はＤＭＳ（Ａ）が結果（Ｒ′）をユーザ（Ｕ）に渡す。

【００４５】以上、基本的な構想を説明したので、本発明の一層詳細な説明を以下に記載する。

【００４６】ＤＭＳ認識（ＤＭＳ－Ａｗａｒｅ）環境下におけるドキュメント管理システム（ＤＭＳ）構成

図３は、本発明によるドキュメント管理システム（ＤＭＳ）Ａの構成を示す図である。図３においては、環境はＤＭＳ認識のある環境であることを前提とする。このことは、データの記憶および検索ならびに他操作の場合において、ＤＭＳ（Ａ）との対話のためのプロトコルは一

様であることを意味する。特に、ＤＭＳ（Ａ）は、ＤＭＳ認識のアプリケーション、およびＤＭＳ認識のある保管場所（repository）において、ＤＭＳプロトコルおよびコードが使用できるようにするするため、ＤＭＳの構成を拡張することが可能となるように開発された。しかし、本発明によって、以下に詳細に述べるように、例えば、ＤＭＳ非認識アプリケーションおよびファイルシステムなど従来から受け継がれてきたシステムと対話可能なことにより新たな利益の付加が実現されることを、発明者らは認識している。

【００４７】図に示すように、ドキュメント管理システム（ＤＭＳ）Ａは、フロントエンド（front-end）コンポーネントＢ、およびバックエンド（back-end）コンポーネントＣとのオペレーション（演算操作）のために接続されている。フロントエンドコンポーネントＢは、ＤＭＳ認識アプリケーション１０ａ～１０ｎ例えばワードプロセッサアプリケーション、メールアプリケーション等を含む。ブラウザ１２（特殊化された形態のアプリケーションと見なされる）も、ＤＭＳ（Ａ）と共に使用するように設計されている。

【００４８】同様に、バックエンドコンポーネントＣは、ドキュメントの内容が記憶される複数の保管場所１４ａ～１４ｎを備えることができる。このような保管場所には、プリンシパルのコンピュータのハードディスク、ファイルシステムサーバ、ウェブページ、動的な実時間処理（リアルタイム）のデータ転送源、および他の保管場所がある。ＤＭＳ（Ａ）は種々の保管場所からデータを受け取ることができ、ビットプロバイダ１６によりデータはＤＭＳ（Ａ）に供給される。

【００４９】プリンシパル１～ｎはそれぞれ、例えばドキュメント２０ａ～２０ｎのようなドキュメントを管理するための自分自身のカーネル１８ａ～１８ｎを有する。ドキュメント２０ａ～２０ｎは、対応するプリンシパル１～ｎがそのドキュメント管理空間にもたらしたドキュメントである。特に、それらは、プリンシパルが価値があると見なしており、このため、そのプリンシパルのドキュメントとして何らかの方法でマークされたドキュメントである。このドキュメントは、例えば、プリンシパルが作成したドキュメントであったり、プリンシパルが送信または受信した電子メール、プリンシパルが見つけたウェブページ、画像の連続した流れを発生する電子カメラなどのリアルタイムの入力データ、またはＤＭＳのドキュメント空間にもたらされた他の何らかの形態の（ビデオ、オーディオ、テキストなどを含む）電子データである。各ドキュメント２０ａ～２０ｎは、配置された静的プロパティ２２および／または動的プロパティ２４を有する。

【００５０】ドキュメント２０ａは、基本ドキュメントであり、参照ドキュメント２０ｂ～２０ｃによって参照される。以下に一層詳細に説明するが、基本ドキュメント２０ａは静的プロパティ２２および／または動的プロパティ２４を有すると共に基本プロパティ２６をも備えている。但しこの基本プロパティ２６は静的プロパティ２２および／または動的プロパティ２４であってもよい。静的プロパティは「－」を付して示されており、動的プロパティは「－○」を付して示されている。

【００５１】参照ドキュメント２０ｂ～２０ｃは、基本ドキュメント２０ａと対話するように構成されている。基本ドキュメントおよび参照ドキュメントは両方とも、静的プロパティ２２および／または動的プロパティ２４を保持することができる。プリンシパル２，３が初めて基本ドキュメント２０ａにアクセスする場合、対応する参照ドキュメント２０ｂ～２０ｃがカーネル１８ｂ～１８ｃのもとでそれぞれ作成される。参照ドキュメント２０ｂ～２０ｃは、それらの基本ドキュメント２０ａをはっきりと識別するためにリンク２８および３０を記憶する。特に、本発明では、各基本ドキュメントはドキュメントＩＤを付けて記憶される。このドキュメントＩＤはそのドキュメント固有の識別子である。参照ドキュメント２０ｂ～２０ｃが作成される場合、それらは基本ドキュメントの特定ドキュメントＩＤへのリンクを発生する。あるいはまた、プリンシパルｎが参照ドキュメント２０ｃを参照する場合、プリンシパル３の参照ドキュメント２０ｃへのリンク３２を備えた参照ドキュメント２０ｎが作成される。このリンクによって（すなわち、そのドキュメントハンドル）、プリンシパルｎは、基本ドキュメント２０ａの基本プロパティおよび共用参照プロパティは無論のこと、プリンシパル３がその参照ドキュメント２０ｃに付加した共用プロパティを見ることができる。このことは連鎖形成の概念を説明している。

【００５２】前述した構成により、プリンシパル間のドキュメントの共有および転送が可能になり、ドキュメントを整理するために必要な柔軟性が提供される。引き続き図３を参照して、この点において、リンク２８～３０は一つのドキュメントから他へのリンクとして示されているが、ＤＭＳ（Ａ）内の通信は一般にカーネル１８ａ～１８ｎ間の通信によって達成されることに注意すべきである。このため、ＤＭＳ（Ａ）とフロントエンドコンポーネントＢまたはバックエンドコンポーネントＣのいずれかとの間の通信、あるいはＤＭＳ（Ａ）内のプリンシパル間の通信が行われる場合、この通信はカーネル１８ａ～１８ｎを介して行われる。しかしながら、本発明は他の通信構成とも同様に動作することを理解されたい。

【００５３】上述した構成を用いる場合、本発明のＤＭＳ（Ａ）は、ファイルまたはフォルダタイプの環境内のような厳密な階層内で操作することをプリンシパルに要求しない。むしろ、ドキュメントに付加されたプロパティ２２，２４によって、プリンシパルが最も有効であることが分かる方法に従って、プリンシパルがドキュメン

トを検索し整理することを可能にする。

【００５４】例えば、プリンシパル１（カーネル１８ａの所有者）が内容付き基本ドキュメントを作成してＤＭＳ（Ａ）内に保管し、プリンシパル２（カーネル１８ｂの所有者）が自分自身の必要性のためにそのドキュメントを使用し整理したいと思う場合、プリンシパル２は参照ドキュメント２０ｂ上にプロパティを配置できる。これらのプロパティを配置することによって、プリンシパル２は、プリンシパル１が考えるのとは異なる手法により基本ドキュメントを検索できる。

【００５５】さらに、ブラウザ１２と対話することによって、プリンシパルは、選択されたプロパティを持つすべてのドキュメントを要求するクエリを実行できる。より具体的には、ユーザは存在するプロパティに対して要求を行うクエリ言語を実行することができる。ブラウザ１２の使用については、以下のセクションにおいて詳細に述べる。

【００５６】従って、本発明のポイントは、特定のプリンシパルに対して適切な動作であってドキュメントの大元の作成者の整理構造やさらに他のプリンシパルとは必ずしも同等である必要がない動作が行われるように、異なるプリンシパルによってプロパティが付加されるドキュメント空間を、ＤＭＳ（Ａ）が管理することである。

【００５７】本発明の別の注目すべき態様は、プロパティを使用すると、ドキュメントに単一のマシン上に存在するように要求する代わりに、ドキュメントの固有の同一性をそのプロパティからすなわちプリンシパルの見方から分離するので、ドキュメントは本質的に複数のマシン上に存在できることである（基本ドキュメント２０ａはカーネル１８ａ～１８ｎのすべてまたはいずれか一つに存在できる）。さらに、ドキュメントに関連するプロパティはプリンシパルが作成したドキュメントに従うので（例えば、カーネル１８ｂのドキュメント２０ｂ上のプロパティは基本ドキュメント２０ａを参照できる）、ドキュメント２０ｂのプロパティが論理的に基本ドキュメント２０ａに関連していたとしても、ドキュメント２０ｂのプロパティはカーネル１８ｂ上で実行できる。すなわち、ドキュメント２０ｂに関連する（基本ドキュメント２０ａを参照する）プロパティがその動作によるいかなる費用をも負担する場合、プロパティをドキュメント上に配置したプリンシパルがプロパティを維持するので、これらの費用はカーネル１８ｂ（すなわち、プリンシパル２）が負担する。

【００５８】図５～８は、本発明の概念を示す図である。既存のファイルシステムの基本理念を図４に示す。より具体的には、ドキュメントＡは、その内容４２によって担持されるシステムの識別情報（すなわち、階層形態）４０を有する既存のシステムを表す。他方、図５は、本発明の概念を示す図であり、プロパティ４４ａ～４４ｎはドキュメントＢの内容４６から分離されていることを示す。このようにドキュメントの内容をドキュメントの記述プロパティから分離することは、ドキュメントの物理的ロケーションに関係なくドキュメントを管理することを意味している。この有利な要因により、プリンシパルは、既存ファイルシステムの厳格かつ階層的な保管場所の要求によらず、プロパティにより、ドキュメントＢと他のドキュメントとを群の形態で関連づけ、ドキュメントを検索し、記憶することができる。

【００５９】たとえば、従来のファイルシステムはドキュメントのロケーションおよび名が共にドキュメントの識別性を構成し、それ故、そのロケーション内にその名と共に現れたいかなるドキュメントもそのドキュメントと認識されていたであろうと推定される。しかしながら、本発明はこれとは異なり、群への帰属関係は可変であり、名は単なるもう一つのプロパティであり、プロパティはドキュメントを見出すための重要な要素である。

【００６０】図５に示される概念の拡張として、図６に注目されたい。第一に、ドキュメントＢは、そのプロパティの一つとして「ＤＭＳに関連するドキュメント」４４ａを含み、別のプロパティとして「１９９８年に作成されたドキュメント」４４ｂを含むと仮定する。そこで、プリンシパルが、すべての「ＤＭＳに関連するドキュメント」群、およびこれとは別に「１９９８年に作成されたドキュメント」群を作成することを望む場合、ドキュメントＢは両群に見出される。これにより、さらに、このプロパティに基づくシステムの明確な態様が示される。より具体的には、対話型の体験に、ドキュメントが同時に複数の位置で出現可能であることが導入される。このことは、ドキュメントは同時に複数の群に含まれることができ、このため２つの群が同時に同じドキュメントを表示可能であることを意味する。

【００６１】図７について述べると、本発明のさらに進んだ概念が図示されており、この概念は、動的プロパティにとって関心のある操作が発生した際における動的プロパティの通知を目的とする。プリンシパル１は、カーネル６０ａによる操作を開始し、その内容が記憶保管場所６０ｃにあるドキュメント６０ｂを検索する。本発明によれば、プリンシパル２およびプリンシパル３は、それぞれドキュメント６０ｂに関して、自身が動的プロパティ６４ａ～６４ｎを付加したドキュメント６２ａ～６２ｎを有することができる。また、幾つかの外部送信元（たとえば、サービス、ページャ、電子メールプロバイダ、など）６６ａ～６６ｎがドキュメント６０ｂに関心を持つこともできる。このモデルによれば、プリンシパル１が操作要求を出すと、通知７０ａ～７０ｎが他のドキュメント６２ａ～６２ｎおよび外部送信元６６ａ～６６ｎに送られる。ドキュメント６２ａ～６２ｎまたは外部送信元６６ａ～６６ｎが、いずれも、この特定の操作要求を考慮して機能するように設計されている場合においては、これらの要素は、次に、自身の機能を実行す

る。たとえば、プロパティ６４ａは「ドキュメント６２ｂがアクセスされる度に通知する」を示し、外部送信元６６ａはドキュメント６２ｂがアクセスされる度に「誰か（Ｊｏｅ）」に電子メールを送ることができる。一旦プリンシパル１がドキュメント６２ｂにアクセスする操作を開始すると、これらの動的プロパティあるいは外部送信元が通知される。

【００６２】発生する可能性がある他の操作として、たとえば、プリンシパル２および３がプロパティを付加したドキュメントをプリンシパル１などの著者が削除することが考えられる。本発明にかかる実施形態においては、この場合にもプロパティは引き続き存在し、プリンシパル２またはプリンシパル３が削除されたドキュメントの内容を検索すると、内容は削除されたという表示が送られる。代替の方法として、このようなドキュメントは削除予定であるという情報を送り、ドキュメントをコピーする機会を与えるという方法がある。無論これ以外の代替方法も可能である。

【００６３】図８は、操作（オペレーション）の概念をさらに拡張し、プロパティの開始に関するトリガイベント（きっかけとなる事象）として動作する概念を示す。コンピュータシステムは、定義済み操作を有する。定義済み操作には、内容読み取り操作７６ａ、編集操作７６ｂ、ビュー操作７６ｃ、保存操作７６ｄ、および他の公知の定義された操作がある。操作とプロパティ例えば静的プロパティ２２および動的プロパティ２４等との対話は、一つの操作は一つ以上のプロパティと関連することができ、各プロパティもまた一つ以上の操作と関連することができることを示す。これは、特定のプロパティを構築する際に、異なる操作へのコールを含むことによって実現される。

【００６４】前述のように、既存のシステムにおいては、応答を異にする領域は厳格に分割されていることが分かる。たとえば、オペレーティングシステムはアプリケーションに由来する明確な応答を有し、ファイルシステムは追加的に定義され内包される応答および能力を有する。たとえば、アプリケーションは、通常、ファイルシステムの操作として定義された操作を実行することはできない。しかし、本発明によれば、アプリケーションは（動的プロパティの形態として）、受け継がれた既存のファイルシステムの保管階層内に通常は内包されていた機能性に対して関わるようになる。より具体的には、動的プロパティは、それ自身が、ある特定の操作の実行に対して関心があること、あるいはそれに対して役に立つことを宣言することができる。これらの動的プロパティは、特定の操作が行われる際に呼び出されるようにコード化される。

【００６５】前述したことは、ドキュメントの識別がドキュメントの内容から分離され、ドキュメントプロパティに基づく場合、ドキュメントの共有、収集、および構成を実行できる方法を述べることを目的としたものである。

【００６６】前述したように、ドキュメント空間との対話は、ドキュメントが整理保管されている構造ではなく、ドキュメントの有意義なプロパティを基礎とする。この様式に則ってドキュメントプロパティを使用することにより、対話が、ユーザの直接関心事および考慮中の作業に対して、非本質的な構造よりもより強固に関連づけされることを意味する。加えて、ドキュメント空間の構造は、ドキュメントが整理保管されたとき、単なるドキュメントの状態のみでなく、ドキュメントの状態の変化をも反映する。しかしながら、収集された群は、その群の内部にも出現し、また例えば、ドキュメント所有権、修正年月日、ファイル形式、などの標準的な整理保管に関する情報は、本発明にかかるシステムによって保存され、インフラストラクチャによって保持されるドキュメントプロパティとして出現する。したがって、プリンシパルは、ドキュメント空間とのより伝統的な形式の構造化された対話を取り戻すことができる。

【００６７】ＤＭＳ非認識コンポーネントを含むドキュメント管理システム（ＤＭＳ）構成

前述したように、図３の概念は、ＤＭＳ認識環境下であるという仮定に基づいて説明した。図９に関する説明は、ＤＭＳがＤＭＳ認識のないコンポーネントと共に使用される状況を対象とする。

【００６８】以下の説明は、発明者によるＪＡＶＡ（ＪＡＶＡおよびＪＡＶＡのついた標章は、Ｓｕｎ Ｍｉｃｏｓｙｓｔｅｍｓの登録商標である）によって認証された実装の観点から述べることに注意されたい。ここでの説明はＪＡＶＡによる本発明の実装に焦点を絞るが、これは本発明をこの言語による本発明の実装に限定することを意図するものではない。言うまでもなく、この実装は多様な異なる言語を用いて行うことができる。また、このＪＡＶＡ実装環境において述べられている説明により、もし異なる環境下において実装される場合には、種々のコンポーネントおよび／または構成は要求されない。たとえば、以下の説明では、三つのインターフェースを述べるが、このようなインターフェースの一つがＪＡＶＡ ＩＯーストリームインターフェースである。もしこの環境以外で記述する場合には、本発明にかかる構成は、二つのインターフェース、すなわちＤＭＳ認識インターフェースとＤＭＳ非認識インターフェースとによって記述可能である。

【００６９】ＤＭＳ（Ａ′）の核（core）は、ドキュメントにドキュメントプロパティを付けるというＤＭＳドキュメント概念を実装するドキュメント層８０である（この層は、たとえば、図３のカーネル、ドキュメント、プロパティ、およびプロバイダなどのコンポーネントを含む）。この実施形態において、ＤＭＳ（Ａ′）は三つのレベルのインターフェースを提供する。第一はＤ

ＭＳドキュメントインターフェース８２ａ、すなわち完全にＤＭＳ認識のあるアプリケーションのためのＪａｖａクラスモデルである。ＤＭＳオブジェクトモデルは、ドキュメントオブジェクト、プロパティ、クエリー、および群によって構成され、プログラマが自身のプログラム中で使用できるクラスのセットとしてプログラマに提供される。これは、ＤＭＳの新規な特徴を活用する新アプリケーションを構築する機構である。

【００７０】第二のインターフェースは、標準ＪａｖａＩＯ－ストリームインターフェース８２ｂであり、ＤＭＳプロトコルを理解しないＪａｖａアプリケーションを統合するためのものである。このインターフェースを用いて、たとえば、ＪａｖａビーンズＢ３を統合し、特定のドキュメントの書式を見て編集することができる。

【００７１】第三のインターフェースは、完全にＤＭＳ非対応である既製のアプリケーション８４のためのトランスレータ（翻訳機）８２ｃである。このようなトランスレータが実装されている例は、ＤＭＳトランスレータに属するＮｅｔｗｏｒｋ Ｆｉｌｅ Ｓｙｓｔｅｍ（ＮＦＳ）サーバ（商標；Ｓｕｎ Ｍｉｃｒｏｓｙｓｔｅｍｓ、１９８９）であり、ＤＭＳデータベースも標準ファイルシステムとしてアクセスすることができる。アプリケーションは、通常どおり単にファイルシステムの読み取りおよび書き込みを行う。ＮＦＳインターフェースは、背後のＤＭＳドキュメントに対して奉仕する。これによって、既製のアプリケーションをＤＭＳにおいて使用することが可能となるのみでなく、これらの外部アプリケーション中に発生する動作に対して関係のあるドキュメントプロパティ（たとえば、更新年月日）を、ＤＭＳが維持することを可能とする。

【００７２】背景のアプリケーションは、サービスコンポーネント８６と呼ばれることもあるプロパティ発生器によってＤＭＳに統合される。ＤＭＳプロパティ発生器は、そのシステムに情報を導入するアプリケーションであり、内容からプロパティに変えるために、多くの場合、組織化されたファイルを処理する。たとえば、メールサービスは電子メールファイル上で動作し、ＤＭＳドキュメントが電子メールのヘッダからの詳細によってドキュメントプロパティとして注釈されるように電子メールファイルを処理する。プロパティ発生器は、定期的に動作するように（たとえば、深夜、または５分ごとに）、または特定の事象に応じて動作するように（たとえば、ドキュメントの内容が変更されたときは常に）スケジュールされる。

【００７３】ドキュメント自身がＤＭＳ中に存続するかわりに、ＤＭＳはプロパティを単に維持する。また一方ＤＭＳは、ドキュメント保管場所Ｃ′からドキュメント内容を提供する。本発明の実施形態において、これらの保管場所としては、各実装プラットフォーム上の局所的なファイルシステム、ならびにワールドワイドウェブなどがある。アプリケーションがＤＭＳを用いてドキュメント内容を取り出すとき、内容は幾つかの外部保管場所から実際に中継される。一様な表示のため、ビットプロバイダ８８ａ～８８ｎは、適切な記憶プロトコルを翻訳する能力を備える。プロパティは、プロパティのデータベース９２上に配置することができる。この図には、ＤＭＳ非認識ブラウザ９４も示される。

【００７４】実装（Implementation）

発明者らによって実装された限定ＤＭＳは、約５０００行のＪａｖａ１．１コードを含む。限定ＤＭＳは、Ｊａｖａ Ｄｅｔａｂｅｓｅ ｃｏｎｎｅｃｔｉｏｎ（ＪＤＢＣ）を用いて、いかなるＳＱＬ（商標）データベースバックエンド（プロパティのための）とも会話し、ＤＭＳは、Ｏｒａｃｌｅ（商標）バックエンドを有する、ウィンドウズＮＴ（商標）を実行するＰＣ（パーソナルコンピュータ）上、およびバックエンドとしてパブリックドメインＭｙＳＱＬ（商標）を使用するＳｏｌａｒｉｓ（商標）を実行するＳｕｎワークステーション（商標）上において容易に実行される。ユーザインターフェースは、Ｊａｖａ Ｆｏｕｎｄａｔｉｏｎ Ｃｌａｓｓｅｓの実装としてＪａｖａＳｏｆｔよりプレリリースされたＳｗｉｎｇ（商標）を用いて実装される。Ｓｗｉｎｇは純粋Ｊａｖａの実装であるので、このインターフェースは完全にクロスプラットフォームである。

【００７５】以下のセクションでは、図９のＤＭＳ（Ａ′）についてさらに詳細に述べる。このシステムのコンポーネントは、二つに分類される。すなわち、使用および対話のＤＭＳモデルを支援することを主目的とするコンポーネントと、従来より受け継がれた環境へのＤＭＳシステムの統合を支援することを主目的とするコンポーネントである。この分割は、完全なものではないが、設計要素の背後にある意図を分類するのに役立つ。しかし、両態様は、ＤＭＳから求められる対話のスタイルを支援するために必要とされる。

【００７６】対話に対する支援

ＤＭＳ導入の背後にある基本的な動機付けは、大量のドキュメント空間との新形式の対話を支援するという欲求である。この新しい手法は、ユーザにとって有意義であり、かつドキュメント分類、編成、探索、および検索のための送信元である高レベルのドキュメントプロパティを、基礎とするものである。

【００７７】対話のスタイルによって、設計上の多数の基準が設定される。第一は性能であり、その理由は、すべてのドキュメント動作はプロパティによって管理されるので、プロパティ管理は直接操作によって対話を支援するために十分な程度に高速であることを必要とするためである。第二は一貫性であり、焦点を個別ドキュメントに置くのみでなく、ドキュメントのセットの全域に置く必要がある。第三基準は知覚安定性（perceptual stability）であり、属性、したがってドキュメント群は

絶え間のない変化を受けるが、自身の足元で絶えず変化するシステムは誰も使用できない。

【００７８】これらの基準は、ドキュメントプロパティを扱うＤＭＳ構成のこれらのコンポーネントの設計に反映される。

【００７９】ドキュメント層

ＤＭＳドキュメント層８０は、付加される任意のプロパティをモデルのドキュメントに付与する。前述したように、ドキュメント層８０自身はドキュメントを記憶しない。その代わり、ドキュメントは、たとえば、標準ファイルシステムおよびワールドワイドウェブなどの様々な既存の保管場所Ｃ′に保持される。ドキュメント層は、３つの機能を有する。

【００８０】１．ドキュメント層は、単独のドキュメントモデルを有するこれらの種々のバックエンドの保管場所へのアクセスを単一化する。

【００８１】２．ドキュメント層は、ドキュメント属性機構を導入し、ドキュメントプロパティを付加、除去、および探索する手段を与える。

【００８２】３．ドキュメント層は、それ自身ドキュメントプロパティに基づく単一化されたドキュメント収集機能を付加する。

【００８３】ドキュメント層８０は、バックエンドデータベースサービスを使用し、ドキュメントプロパティを記録する。既存の実装においては、このデータベースサービスは、ＤＭＳコードが使用中の特定のデータベースプロダクトと独立となるように、Ｊａｖａデータベースコネクション（ＪＤＢＣ）を介して伝達される。任意のプロパティをドキュメントと関連づけることができる。静的プロパティは、簡単な名と値の対である。多数の静的プロパティは簡単な文字列または文字列リストであるが、属性値はシリアライズされたＪａｖａオブジェクトとして記憶されるので、任意の複雑なデータ構造をドキュメント静的プロパティとして記録することができる。動的プロパティは、幾つかの形式の動作をドキュメント上にてまたはドキュメントに関連して実行する点で異なる。

【００８４】ドキュメント群

個別ドキュメントの周囲に編成されるドキュメントシステムは、最適なシステムであっても、使用するのに時間がかかる。ＤＭＳ（Ａ′）中の大部分の対話は、ドキュメント群の要素としてのドキュメントとの対話である。ファイルシステムドキュメントおよびウェブドキュメントと共に、ドキュメント群は、ドキュメント形式として実装されるので、これらのドキュメントは、ドキュメントに適用可能な操作と全く同じ操作を受ける（関連するプロパティを持つこと、探索および検索、ならびに自身が群のメンバであることなど）。

【００８５】本発明の実施形態よれば、ドキュメント群は、三つの要素を含む（そのそれぞれは空文字とすることができる）。第一は、クエリー項目（query term）である。クエリー項目は、ドキュメントプロパティの表現により特定される。クエリーは、ドキュメント上の特定のプロパティの存否について試験すること、プロパティの特定値を試験すること、または形式特有値の比較を実行することができる（たとえば、「２時間以内に変更」および「先週修正」など広範囲のデータ指定を与えることができる）。ドキュメント群中のクエリー項目は「存続する」。群は、いかなるときにも整合ドキュメントを含むので、ドキュメントはそれ自身の即時状態によって出現または消失する。

【００８６】クエリー項目に加えて、ドキュメント群は、包含リストおよび排他リストと呼ばれる、ドキュメントの２つのリストを記憶する。包含リスト中のドキュメントは、ドキュメントがクエリーと一致するか否かに関係なく群のメンバとして戻される。排他リスト中のドキュメントは、ドキュメントがクエリーと一致する場合でも群のメンバとして戻されない。クエリーが空文字である場合には、包含リストが群の内容を有効に決定する。

【００８７】したがって、いかなるときにおいても、群の内容は、包含リストのドキュメントに、クエリーに一致するドキュメントを加えて、排他リストのドキュメントを差し引いたものである。これらの三部構造を「流動群」と呼ぶこととする。ドキュメント群におけるこの実装の目標は、普通の形式のドキュメント編成および検索を支援することである。クエリーを用いて、最初の群を作成すること、またはデフォルトの帰属関係を特定することができる。しかしながら、帰属関係は、クエリーを再編成する必要なく、ドキュメント群の内容の直接的な操作によって、さらに精密化することができる。項目の追加および削除によりクエリーの結果をオーバーライドすることが可能であり、これらの変化は永続する。ブラウザもクエリー項目の直接操作を支援するので、クエリーの再編成はかなり簡単な操作である。

【００８８】群、クエリー、およびプロパティは、ＤＭＳドキュメント空間とのすべて対話の基礎であり、したがって、プロパティエンジンの性能は、ＤＭＳシステムにおける重要な要素である。ＤＭＳデータベースエンジンは対話応答に必要な性能条件を十分に満足する。小型の試験データベース（３４２ドキュメント、４９１１属性）によって、クエリー"メール発信元＝ｄｏｕｒｓｈ"を評価した結果、８ドキュメントを戻すために３０ｍｓ（ミリ秒）を要し、一方、クエリー"ＭＩＮＥＴｙｐｅ＝ｔｅｘｔｈｔｍｌ　または　ＭＩＮＥ　Ｔｙｐｅ＝ｔｅｘｔｊａｖａ　かつ　１月以内に読まれたもの"では３２ドキュメントを戻すために１４０ｍｓを要した。同じクエリーを大型データベース（２５５８ドキュメント、２７９２１プロパティ）において実行した際には、９０ｍｓ（８ドキュメント）および６２０ｍｓ（３００

ドキュメント）をそれぞれ要した。

【００８９】プロパティ発生器

ドキュメントプロパティによって編成されたドキュメント保管場所は、ドキュメントが実際にプロパティを有する場合にのみ有用である。ドキュメント上には幾つかのプロパティの送信元が存在する。

【００９０】第一に、プロパティはプリンシパルに由来する場合があり、プリンシパルは任意のプロパティをドキュメントに付加し、その結果、自身の構造を作成することができる。実際、このシステムの目標は、プリンシパルの作業に関連のあるプロパティのセットを作成することによって、プリンシパルがプリンシパル自身の構造を作成し、次に、それらの構造を用いてドキュメントを編成し、検索することを可能とすることである。第二に、プロパティは、動的プロパティ、アプリケーション、サービスの少なくともいずれか一つによっても作成される。

【００９１】しかし、重要なプロパティはドキュメント内容から導くことができるので、別の機構によって、プロパティによりドキュメントに自動的にタグを付ける手段が提供される。たとえば、ドキュメントの形式、長さ、作成者、作成年月日、など、幾つかのドキュメントプロパティは汎用であり、これらは、明らかにＤＭＳが直接に維持するドキュメントプロパティである。他の関係のあるプロパティは、内容に対して特有である場合がある。たとえば、電子メールメッセージは、そのヘッダ内容に関する情報によってタグを付けることができ、ＨＴＭＬファイルはそのヘッダからの情報、またはそれに含まれる他のドキュメントリンクによってタグを付けることができる。この機能は、プロパティ発生器によって実現できる。

【００９２】図１０は、たとえば、ブラウザ９６ａを操作するユーザを介し、あるいは特殊なアプリケーション９６ｂにより、あるいは動的プロパティ９６ｃによりプロパティを付加する方法を示す図である。さらに、プロパティは、例えば、ＨＴＭＬプロパティ発生器９６ｄ、電子メールプロパティ発生器９６ｅ、または画像プロパティ発生器９６ｆ等のプロパティ発生器により付加することができる。以下の説明から分かるように、図にはプロパティ発生器の代表的な例のみを示すことを理解されたい。プロパティ発生器は、この方式によってファイルを解析するために使用できる一遍のコードとすることができる。プロパティ発生器は、前述したように、たとえば、電子メールメッセージおよびＨＴＭＬドキュメントなどの汎用構造ファイル形式を構成している。ＤＭＳ（Ａ′）も、さらに特殊化した複雑な発生器を形成する。その一例として、Ｊａｖａ送信元ファイルを解析し、ドキュメント上のプロパティに関するパッケージ、インポート、および方法定義についての情報をコード化することができるＪａｖａサービスがある。また、特定の特殊化プロパティ発生器として、システムの外部に存在する他の数編のソフトウェアのための発生器包（generater wrapper）が備えられる。例えば、ドキュメント集計ツールは発生器モードによってＤＭＳ（Ａ′）に組み込み済みであるので、ドキュメント内容は、ドキュメントプロパティとして利用可能となったキーワードと文とを用いて集計される。

【００９３】プロパティ発生機構は、アプリケーション特有のＤＭＳ空間を構築する経路でもある。その一例として、夏期教育実習生のデータベースのフォーマットを理解するプロセッサがあり、各アプリケーション記録ドキュメントは、教育実習生の熟練度および関心事、実習生の学校、程度、関心のある話題に関する情報などによってタグを付けることができる。次に、ＤＭＳ（Ａ′）を用いて、教育実習生を解析し編成することができる。

【００９４】このようなプロパティ発生器は、所定のドキュメントに対して利用できるプロパティの数を増加することによって、ドキュメント空間との対話の質を向上させる。発生器を用いてドキュメントからプロパティを抽出することによって、システムは内容情報を抽出し、それをプロパティ中心のドキュメント構造によりコード化し、内容に基づく手法と構造的手法とを橋渡しする。

【００９５】ＤＭＳ（Ａ′）は発生器によってこのリンクを形成するので、発生器がドキュメント内容の変更に応答できることが重要である。プロパティ発生器は、情報を更新して維持するために種々の時点において実行されるようにスケジュールを決めることができる。プロパティ発生器は、１日の特定の時刻に（たとえば、午前４時に主要処理を実行するように）、特定の間隔で（たとえば、１０分ごとに）、または特定の事象発生時に（たとえば、ドキュメントにプロパティが付加されたとき、またはドキュメントが書き込まれたとき）に実行することができる。

【００９６】統合化に対する支援

日常世界における実際的な観点から、ドキュメント管理システムは統合化され、拡張性を有する必要がある。結局、ＤＭＳ（Ａ′）は既存のドキュメント空間の編成および探索を支援することを目的とするにも関わらず、既存のドキュメント空間は多種多様なフォーマット、構想、およびアプリケーションを使用している。

【００９７】最初の動機付けであった対話のモデルによって新形式のドキュメント対話が生まれ、新形式のドキュメント対話は新アプリケーションによって明確に具体化され、またＤＭＳ（Ａ′）が必ず提供する種類の特徴を利用することができる。しかし、同時に、本発明における既存のアプリケーションに対する支援は大いに必要とされる。

【００９８】既存のアプリケーションを適応させるため、ならびに新しいアプリケーションの開発を準備するため、ＤＭＳ（Ａ′）は、事前に導入される三つのアプ

リケーションインターフェースを提供する。以下の注記は、既存の実装に関してその導入について詳述する。

【００９９】本来のアプリケーションに対する支援

ＤＭＳドキュメントインターフェース８２ａは、Ｊａｖａオブジェクトとしてドキュメントへのアクセスを提供する。アプリケーションは、関連したパッケージをそれらのＪａｖａコードにインポートし、また、ドキュメント、群、およびプロパティにアクセスするために設けられたＡＰＩ（アプリケーションプログラムインターフェース）へコーディングすることにより、このインターフェースを利用することができる。これは、新しいＤＭＳ認識アプリケーションを構築し、新しい対話モデルの実験をするための標準的な手段である。（図３の）ＤＭＳのブラウザ１２は、ＤＭＳアプリケーションと見なすことができ、このレベルにおいて構築される。ＤＭＳのドキュメントインターフェース８２ａは、本願で説明する機能性（例えば、群、WWWドキュメントへのアクセスなど）を支援する特定のサブクラスを有して、ドキュメントおよびプロパティのクラスを提供する。アプリケーションは、例えば内容特定の可視化を用いて、ＤＭＳドキュメントの直接視を可能とし、またはプロパティベースのドキュメントサービスのバックエンドとしてＤＭＳを利用する全体的に異なるインターフェースの提供を可能とする。

【０１００】第二に、ＤＭＳドキュメントへのアクセスは、Ｊａｖａ ＩＯストリームインターフェース８２ｂによって提供される。ＤＭＳ ＩＯストリームは、標準Ｊａｖａストリームモデルのサブクラスに属するので、いかなる標準Ｊａｖａアプリケーションに対してもＤＭＳの機能を利用可能とする。本発明の実装においては、新しいアプリケーションを開始するオーバーヘッドなしにドキュメント内容へのアクセスを提供することができるモデルを使用し、たとえば、画像およびＨＴＭＬファイルなどのＪａｖａビーンズを組み込んだ。

【０１０１】既製のアプリケーションに対する支援

第三レベルのアクセスは、トランスレータ８２ｃ（ＮＦＳプロトコルを実装するサーバ）による。これは、純粋Ｊａｖａにおける本来のＮＦＳサーバの実装である。このトランスレータ８２ｃ（すなわち、ＤＭＳ ＮＦＳサーバ）は、すべてのＮＦＳのクライアントに対してＤＭＳのドキュメント空間へのアクセスを提供する。すなわち、サーバは例えばマイクロソフト ワード（商標）など現在市販されているアプリケーションがＤＭＳドキュメントを利用できるようにすること、またＰＣ上では、ＤＭＳはこれらのアプリケーションにとっては単に別のディスクのように見えるが、ＵＮＩＸ（商標）のマシン上では、ＤＭＳ（Ａ′）は標準的なネットワークのファイルシステムのように見える。

【０１０２】大事なことであるが、このトランスレータを経由して達成されることは、ＤＭＳ（Ａ′）が既存または既製のアプリケーションに対して、内容およびプロパティの読取り／書込み経路内に直接存在することである。別の方法は、ワードなどのアプリケーションによって、ＤＭＳ（Ａ′）に適応するために変更できないような従来のファイルシステムに書き込まれたファイルを後処理しようとすることである。そのかわり、これらのアプリケーションに対してファイルシステムのインターフェースを直接的に提供することにより、内容およびプロパティの読取り／書込み経路上で関連するプロパティを実行することができる。さらに、関連するプロパティ（ドキュメントがいつ最後に使用または変更されたかを記録するプロパティ）は、継続して更新されることが保証されている。アプリケーションがファイルシステムの情報を使用するように書き込まれた場合にあっても、ＤＭＳ（Ａ′）はファイルシステムであるため、ＤＭＳのデータベースは更新される。

【０１０３】ＤＭＳのデータベース層に対するそのインターフェースの部分として、ＮＦＳはクエリ機構へのアクセスを提供する。適切にフォーマットされたディレクトリ名は、クエリとして解釈され、これらのディレクトリ名はクエリによって戻されたドキュメントを「含んでいる」ように見える。ＤＭＳはこのＮＦＳサービスを提供するが、ＤＭＳは保管層ではない。ドキュメントは実際に別の保管場所にある。しかしながら、ＮＦＳ層の使用により、各種の他の保管場所へのアクセスが均一になる（そのため、ウェブ上で利用可能なドキュメントは、ネットワーク化されたファイルシステム内のドキュメントと同じ空間内にあるように見える）。この均一性ならびに読取りおよび書込み経路内における存在によるドキュメントプロパティ随時更新能力の組み合わせにより、ＮＦＳサービスは、公知のアプリケーションとの統合の所望のレベルに対して価値のあるコンポーネントとなる。ＮＦＳプロトコルを実行するサーバ以外に他のサーバも使用できることは理解されよう。

【０１０４】プロパティ

最初に、静的プロパティと、静的および動的プロパティに共通のプロパティの態様とについて、幅広く説明する。次に、動的プロパティに直接に関連する事項を述べる。

【０１０５】最も簡単なプロパティは、ドキュメントのタグである。たとえば、「重要」または「Ｋａｒｉｎと共用」は、ドキュメントのユーザに関するドキュメントのファセット（facet）を表すタグである。ごく僅かに複雑なプロパティは、名と値の対である。たとえば、″作者＝ｋｅｄｗａｒｄｓ″はプロパティであり、その名の成分は″作者″であり、値の成分は″ｋｅｄｗａｒｄｓ″である。本発明によるこれらのプロパティについて、注意すべきことが二点ある。第一は、同じ名を有する複数のプロパティが存在できることである。ドキュメントの著者が複数である場合、ドキュメントは複数の著者プロ

パティを持つ。第二は、プロパティの値は任意のデータとすることができることである。実施例においては、通常、簡単な試験文字列を用いるが、実際は、任意のデータ、またはコードさえも、本発明の実装によってプロパティ値として記憶することができる。

【０１０６】ユーザが実際に見て操作する大量のプロパティを考慮するが、これらの静的プロパティは、プロパティ機構の一成分を構成するに過ぎない。もう一つは動的プロパティである。

【０１０７】静的プロパティ

プロパティは、基本ドキュメントに直接に関連するか、またはプリンシパルに関連するドキュメント基準に分類されるかのいずれかである。基本ドキュメントに関連するプロパティは、基本プロパティであり、「公開（published）」である。公開プロパティの目的は、たとえば、ドキュメントのサイズまたは内容の種類など、所定のドキュメントに固有の情報を表すことである。したがって、基本ドキュメントへのアクセスを有するプリンシパルは、公開プロパティを見ることまたは検討することができる。それ故、ユーザは、個人用情報について公開プロパティを使用してはならない。たとえば、プリンシパルが用いたプロパティがプロパティ"interesting（おもしろい）"である場合（すなわち、ユーザは、"interesting"として定義されるプロパティを有するタグをユーザが付けたすべてのドキュメントを収集することを望む）、このようなプロパティが固有のものであることは滅多にない。

【０１０８】ドキュメントまたは参照のプロパティは、それ自身階層構造とすることができる。すなわち、プロパティは、サブプロパティを有することができる。サブプロパティは、親プロパティに付加しなければならないので、親プロパティは、サブプロパティが付加される可能性がある前に明確に作成されねばならない。親プロパティは、その上、明示的に削除実行する必要がある。最後の子のプロパティを除去することによっても、親プロパティは自動的には除去されない。親プロパティの存在を強制することによって、一度に階層の一レベルを列挙する一様な方法が保証される（すなわち、gerSub=Properties()によっては、次のレベルのプロパティにのみ戻ることができる）。

【０１０９】各プロパティは、名を有する。このことは、階層名を用いて階層を横断することができることを意味する。例えば、「Get me the 'from' sub-property of the 'mail' property of this reference to Grandma's cookie recipe（祖母のクッキーレシピに対するこの参照の＜メール＞プロパティの＜差出人＞サブプロパティを取得せよ）」は、その参照において開始され、「メール」と名付けられたプロパティを見出し、「差出人」と名付けられたそのサブプロパティを見出す。

【０１１０】いかなるレベルの階層も、同じ名を有する複数のプロパティを有することができる。たとえば、プリンシパルが、"作者＝ｊｏｈｎ"および"作者＝ｊｏｅ"の両者を同じドキュメントに付加する場合、それぞれがそれ自身の著者についてさらに詳しく記述したサブプロパティを有する。いずれのプロパティに対するクエリーも、目標とするドキュメントを識別する。プリンシパルはプロパティの値を要求する場合、幾つかの方法の内の一つを用いることができる。標準的なgetValue()は、単独の値を戻し、このような値が存在しない場合または複数存在する場合は、例外を投げ返す。他の変形によって、複数の値が存在する場合でも、一つの値を戻すこと、またはすべての値を戻すことができる。

【０１１１】静的プロパティの値は、直列可能Ｊａｖａオブジェクト（またはｎｕｌｌ）であることが可能である。いずれのプロパティ値についても、分類保証はされない。プリンシパルは、規則に従って、プロパティによって、互換性のある型を記憶し検索しなければならない。プロパティは、任意の値を含むことができるが、プリンシパルは、プロパティのサイズを小さく維持することに努める。大きなプロパティ値は、恐らく、参照として他のドキュメントに記憶されるべきである。Ｊａｖａオブジェクトの文字列表現はプロパティの値であり、デフォルトによる値によってプロパティを探索するとき使用される。

【０１１２】プロパティの可視性は、プロパティの適用者によって位置決めされるタグによって実現される。タグの値は、私用または共用とすることができる。私用プロパティは、著者以外のいかなるプリンシパルも見ることができない。共用プロパティは、要求するいかなるプリンシパルも見ることができる。したがって、著者プリンシパルがプロパティのセットを要求する場合、すべての共用プロパティは戻されるが、私用プロパティは全く戻されない。基本ドキュメント上のすべての基本プロパティは共用と標識付けされるので、公開プロパティはすべてのユーザが見ることができる。

【０１１３】プロパティがドキュメントに付加されるとき、付加者の識別がプロパティと共に記録される。複数のプリンシパルが同じプロパティ（同じ値を有する）を付加し、あるプリンシパルがプロパティを除去することを決める場合は、そのプリンシパルが以前に付加したプロパティのみが除去される。この手法は、プロパティの一つのコピーのみによってドキュメントにタグを付け、そこで、第二プリンシパルはプロパティを維持することを意図するときに第一プリンシパルに誤ってそのプロパティを除去させることのないように取るべき方法としての機能を果たす。

【０１１４】サブプロパティに加えて、各プロパティは、プロパティについて記録される固定されたセットの属性も有する。これらの属性は、プロパティ上のプロパティと考えられる。但し、プロパティ単位の属性は拡張

可能ではない。前述した私用および共用タグならびにプロパティの付加者の識別は、プロパティ単位の属性の例である。他の例としては、そのプロパティが付加された時、およびそのプロパティをブラウザによって表示すべきかどうか、がある。

【０１１５】図１１は、プロパティ"coolness=high"をＤＭＳ（A'）に関係するすべてのドキュメントに付加して、プロパティが設定されたすべてのドキュメントを含む群を作成するプログラムについて、Ｊａｖａでコード化した例を示す。第一ステップは、データベースを開始することである。ＤＭＳ（A'）をセットアップし、データベースをローカルに管理すること（アプリケーションのアドレス空間内で）、またはリモートＤＭＳ構成に接続することができる。この例では、要求はローカルで処理する。DMS.startMysqlDatabase()へのコールによって、ＭｙＳＱＬデータベースに関するデータベースオブジェクトが開始される。次に、クエリー構成子に探索項を付与することによって要求されているドキュメントに対して、クエリーが構成される。クエリーは、群内に密閉される必要はない。クエリーは、find()メソッドを用いて直接に評価することができる。これによって、それと対比してクエリーを評価すべきドキュメントのリストを特定する論拠を得ることができる。この場合のように、リストが省略されると、次に、クエリーは全データベースについて実行される。クエリーは、一致するドキュメントを含むDocListオブジェクトに戻る。ドキュメントを一つずつ処理し、所望のプロパティを設定するため、これらのドキュメントに関する列挙（enumeration）を獲得する場合がある。

【０１１６】ここで、動的クエリーを含む新しい群が作成される。メソッドcreateCollection()によって、新しい群が作成され、戻される。群は、次に、命名される。名プロパティは、実際に"ＤＭＳ"であるが、多くの共通のＤＭＳプロパティはクラスＤＭＳＩｔｅｍに静的に形成される。クエリーは、setQuery()メソッドを用いて群のために設定され、次に、一致するドキュメントの数がプリントされる。このとき、クエリーは群内に密封されているので、データベース内で永続し、次回、ＤＭＳアプリケーションが開始されるときまで存在する。最後に、終了で終わる。

【０１１７】前述した説明を要約して、プロパティの重要な態様を以下に述べる。

【０１１８】・プロパティは、階層内の各ドキュメントのすぐ下に記憶することができる。

【０１１９】・プロパティ階層と隠れプロパティとの組合せにより、名が空白となっている群の問題が解決され、大きなプロパティのセットが管理される。

【０１２０】・同じ名を有するプロパティを一つのドキュメントに複数回付加することができる。そのプロパティ上の種々のgetValueメソッドによって一つ以上の利用できる値が存在するとき、その値の一つ、すべての値、または単独の値を通知と共に戻すことができる（一つを予測しただけのとき）。

・それぞれがそれら自身のプロパティである複数のプロパティ値を作成することによって、複数の値によるクエリー実行は、一つだけのプロパティによるクエリー実行と同じになる。

【０１２１】・プロパティは、私用または共用のタグを付けることができる。私用は、所有者以外のものはアクセスすることができない。共用は、ドキュメントへのアクセスを有するものは誰でもアクセスすることができる。

【０１２２】・所望により、プロパティは、よりきめの細かいアクセス管理を強制することができる。

【０１２３】・クエリー言語によって、プリンシパルは、どのドキュメント上のどのプロパティにプリンシパルが関心があるかを特定することが可能となる。

【０１２４】・プロパティ値は、任意のシリアライズ可能なＪａｖａオブジェクトである。

【０１２５】・各プロパティによって、システムは各プロパティがドキュメント上にある根拠、たとえば、誰がプロパティをそこに位置決めしたかを記憶する。そして、プロパティが一つ以上の根拠についてその存在を明らかにされ、その後一つの根拠が除去される場合、適切な事情のプロパティのみが除去され、ドキュメントは他の根拠によるプロパティを保持する。

【０１２６】・独立のプリンシパルは、他のプリンシパルがそのプロパティを使用することを無視しながら、プロパティを位置決めすることおよび除去することができる。

【０１２７】動的プロパティ

前述した静的プロパティは、データをドキュメントに付加する。静的プロパティは、その後、探索または検索することができる情報を記録する。しかし、ドキュメントの幾つかのプロパティは、ユーザがドキュメントと対話すべき手法に関する因果関係、および対話中のこれらのドキュメントの挙動に関する因果関係を有する。ここでプロパティ「私用」について考察する。単にドキュメントを私用として標識付けすることは、一般に、ドキュメントが他者によって読み取られないことを保証するには十分でない。したがって、「私用」プロパティはタグを超えるものでなければならない。「私用」プロパティは、どのようにしてドキュメントにアクセスするかを管理する手段でもあるべきである。

【０１２８】動的プロパティ機構により、たとえば、「私用」のようなプロパティによって要求されるような挙動を生じ、ドキュメントの状態のみでなく、その挙動にも影響を及ぼす手段が備えられる。同時に、動的プロパティは、ドキュメントを中心とし、ユーザに有意義であり、かつドキュメント消費者によって管理される方式

の対話管理を、プロパティを基礎とするシステムの利点を維持する手法によって可能にする。

【０１２９】動的プロパティは、静的プロパティと全く同様にドキュメントに付加することができるが、動的プロパティは、ドキュメント操作の実行に関係のあるプログラムコードも含む。動的プロパティは、図７に関連して前述したように、操作が行われるときに通知されうる。動的プロパティは、第一の位置においてこれらの操作の妥当性検査に関係することもできる。または、動的プロパティは、操作の実行に関係することができる。これらの各点において、動的プロパティは、プログラムコードを実行することができる。たとえば、通知を使用し、共同システムにおける並行作業の相互認識を維持することができる。通知によって、プロパティが、所定のドキュメント上で操作の周辺を探索し、操作を記録し、ユーザインターフェースによって操作を見えるようにする手段が備えられる。検証を用いて、たとえば、前述した「私用」プロパティなどの機構を実装することができる。これらの機構は、通常、ドキュメントの所有者以外のだれかから発信される読み取り要求の妥当性検査を拒否する。また、操作の実行を支援する一連のプロパティを用いて、たとえば、暗号化および圧縮などの機能をドキュメント上のプロパティとして提供することができる。

【０１３０】動的プロパティは、コードと関連付けられていることによって作動されるプロパティである。特に、プロパティは、Ｊａｖａクラスと関連付けることができる。各プロパティは、それ自身のクラスを有することができるが、一実施形態においては、同じタイプのプロパティは、クラスを共有することができる。このクラスは、ドキュメント上の種々の操作に対応するメソッドを含む。たとえば、プロパティは、基本ドキュメントによって提供される読み取り操作の頂部に「積み重ねられる」プロパティ自身の読み取りメソッドを形成することができる。

【０１３１】完全なオブジェクトとして、プロパティインスタンスオブジェクトは、複数の方法で対話し、その動作を実行することができる。前述した、通知、妥当性検査、および実行に関するこれらの標準機構は、上下に拡張され、その場合、標準機構自身は、下記に関係し、それを含むことができる。

【０１３２】・標準機構は、プロパティ上の情報を付加、除去または変更する試行の打診を受けることができる。これにより、標準機構が、プロパティ情報の妥当性を検査し、その情報の効果を達成するために必要とされる初期設定を実行し、その情報が除去されるときこのような動作を中止することが可能となる。

【０１３３】・標準機構は、自身のドキュメント上の種々の操作の中断を要求することができる。これにより、標準機構が、自身のドキュメントの挙動または自身のドキュメントの他のプロパティの挙動をモニタまたは変更することが可能となる。

【０１３４】・標準機構は、自身のまたは他のドキュメント空間内の動作の通知を要求することができる。これによって、標準機構が、たとえば、付与または推論されるプロパティの更新などドキュメントにまたがる情報を維持することが可能となる。

【０１３５】・標準機構が実装するＡＰＩは、ドキュメント空間の内側または外側のいずれかの他の実体がアクセスすることができる。これによって、標準機構は、自身のドキュメントの基本ＡＰＩを効果的に拡張することが可能となる。

【０１３６】・標準機構は、それ自身または他のドキュメントあるいはそれらのプロパティのＡＰＩを呼び出すことができる。これによって、幾つかのドキュメントに関係する挙動が可能となる。

【０１３７】すべての動的プロパティは、名、値、および動的メソッドという３つの必須の特質を有する。したがって、いかなるプロパティでも、それに動的メソッドを付与することによって動的にすることができる。静的であると考えられていたプロパティでも、それらのgetValueおよびsetValueメソッドはそれらのクラスオブジェクトによって形成されるので、幾つかの面で動的である。その動的メソッドによりプロパティの値を用いて、そのプロパティに関連する永続するデータを記憶することができる。

【０１３８】ドキュメント上で実行できる所定の操作の場合、動的プロパティは、３つのメソッドまで高めることができる。第一は、（ブール：boolean）妥当性検査メソッドであり、第二は、（オブジェクト：Object）主メソッドであり、第三は、（ボイド：void）通知メソッドである。操作が実行予定であるとき、カーネルは、第一に付加されたプロパティ上においてその操作について定義された妥当性検査メソッドを実行する。メソッドが偽（false）で戻る場合、実行は停止する。それ以外の場合は、主メソッドが、定義された順序規則に従って実行される。最後に、すべての通知メソッドが実行される。

【０１３９】次のパラグラフ（ｉ～ｘｉ）において、動的プロパティの特徴を述べる。

【０１４０】（ｉ）プロパティ状態記憶

・プロパティ状態を、プロパティの値として、動的プロパティのサブプロパティとして、同じドキュメント上の他のプロパティとして、別のドキュメントして、または外部記憶システムに、記憶することができる。（プロパティ値の使用およびサブプロパティは助成されるが、）この決定には、プロパティ作成者が適任である。

【０１４１】・プロパティの特別な記憶機構はどのようなものでも複雑になる。

【０１４２】（ii）動的プロパティは、オブジェクトイ

ンスタンスであり、短命である。

【０１４３】・プロパティのインスタンスオブジェクトは、動的プロパティのコードの実行中は存在する必要があるが、それ以外には、カーネルがこのオブジェクトの便利さを見出した時にのみ存在する。カーネルは、オブジェクトをほぼ無限に保持する場合から、何らかの動作を実行するため動的プロパティが想定されるときのみ動的プロパティを作成して動作の完了時にそのオブジェクトを放棄する場合まで、その方針を変えることができる。どのプロパティインスタンスに対しても、現存するオブジェクトの一つのインスタンスが存在する。

【０１４４】・動作をＪａｖａオブジェクト内に位置決めすることは、動作をスレッド（thread）に投入するより負担が軽い。動作を短命なオブジェクトに投入すると、さらに負担が軽く、その上、確実に、プロパティが依存するすべての状態をプロパティ値として見ることが可能となる。これによって、プロパティ中の探索可能な情報と、それらの動作とを明確に分離することができる。この設計は、「プロパティインプリメンタは、インプリメンタのプロパティまたは他のプロパティ（またはドキュメント）中にあるとき以外は実行中の状態を保持することができないので、より困難な作業をしなければならない場合がある。」ということを意味する。

【０１４５】（iii）プロパティに関する状態はそのオブジェクトによって管理される。

【０１４６】・オブジェクトは、プロパティ事例を付加、除去、または変更する試行を検査するために呼び出される。プロパティを付加する場合、オブジェクトは情報によって構成され、付加が妥当であることを検査するaddSelf()が呼び出され、適切な初期設定を実行する。そこで、付加が適切であるかを言及する状態に戻り、許容されない場合は、プロパティはドキュメントに付加されない。同様に、removeSelf()は、プロパティを除去する試行のときに呼び出され、changeSelf(PropertynewProp)はプロパティの値を変更する試行のときに呼び出される。ここでnewPropは新しく記憶される予定の値を表す。

（iv）プロパティ操作には２つの有効範囲、ドキュメントとプロパティとがある。

【０１４７】・ドキュメント有効範囲の操作は、どのプロパティとも独立して実行される。これらの操作は、論理上、document.operation(args)の形式をとる。この形式での操作のために行われる動作は、ドキュメント上のプロパティによって無効とすることができるが、無効操作は既存の操作と同じセットの引数をとる（中断および修正につては下記を参照されたい）。

【０１４８】・プロパティ有効範囲の操作は、プロパティに関して実行される。これらの操作は、論理上、document.property.operation(args)の形をとる。

【０１４９】（v）ドキュメントおよびプロパティはそれらが提供する操作を記述することができる。

【０１５０】・種々の操作により必要とされるパラメータおよび引数の記述が利用可能である。

【０１５１】・プロパティにより実装されるＡＰＩは、そのドキュメントへのアクセスを有する如何なるものからもアクセスすることができる。

【０１５２】・システムおよび他のプロパティのユーザは、最初に、GetDelegateForInterface()を呼び出し、動的プロパティによって実装されたインターフェースを獲得することによって、動的プロパティ操作を呼び出すことができる。

【０１５３】（vi）動作および挙動は、中断および修正されうる。

【０１５４】・プロパティは、登録し、そのドキュメント上の操作に介入することができる。中断することができる基本操作には、内容の読み取りおよび書き込み、ならびにプロパティの付加、変更、および除去などがある。プロパティ操作の中断は、中断するプロパティと同じ参照上の操作に適用される。参照上のプロパティによる内容操作の中断は、参照が永続しているドキュメント空間から生成されたアクセスに適用される。

【０１５５】・プロパティは、それらのドキュメントのプロパティ、またはそのドキュメントについてのそれらのプリンシパルにおける表示態様（view）（またはドキュメントハンドル）であると考えられるため、プロパティがドキュメントに影響を及ぼすことができることは意味をなす。プロパティは、他のドキュメントとの対話を要求する場合、通知機構および種々のＡＰＩを使用することができる。

【０１５６】（vii）プロパティ実行は、妥当性検査、実行および修正段階からなる。

【０１５７】・プロパティは、妥当性検査および通知のためのメソッド、ならびに実行のための主メソッドを帯びる。操作は、その実行前に、付加されたすべてのプロパティによって妥当であると宣言されなければならない。

【０１５８】・メソッド実行をこれらの段階に分割することによって、プロパティ順序づけが簡単になる。妥当性検査によりメソッドは、個別の根拠に基づいて他の実行を無効とすることができる。妥当性検査および通知は、自然クラスの挙動に対応する。

【０１５９】（viii）動的メソッドは順序付けられる。

【０１６０】・ドキュメント上の動的プロパティは、リストまたはベクトルによって順序付けされる。主メソッドの呼び出しの順序は、プロパティの順序に従うが、スタック（stack）形式である。主メソッドを有する第一プロパティは、自身のメソッドを呼び出させ、スタックの残りの主メソッドを同様な様式でプロパティに実行させるハンドルを送る。したがって、スタック中のメソッドは、後続のメソッドのため、引数を変形すること、後

続のメソッドのスタックによって送り戻される結果を改訂すること、または後続のメソッドを全く呼び出さないことができる（妥当性検査または修正メソッドは、互いに影響を及ぼさないので、これらについての順序は問題ではない）。

【０１６１】・これらが、プロパティリスト再順序づけの機構である。

【０１６２】（ix）基本操作は、すべての参照操作の後で実行される。

【０１６３】・基本ドキュメント上の操作は、参照ドキュメント上の操作と結合されるが、プロパティがいつ付加されたかに関係なく、最後に実行される。たとえば、読み取りでは、すべての参照読み取りは、どの基本読み取りよりも前に実行される。BitProviderプロパティは、常時、読み取り／書き込み操作において最終決定権を有する。

【０１６４】（x）プロパティメソッドは、操作を起動させるプリンシパルによってパラメータ化される。

【０１６５】・プリンシパルが操作を要求するための識別子は、各動的プロパティメソッドに送られる。これによって、動的プロパティコードが、誰がその操作を実行しているかに基づいて、その挙動を変更することが可能になる。

【０１６６】・このことは、プロパティに直接に実装されるアクセス管理および通知体系にとって特に重要である。

【０１６７】ブラウザの使用によるプロパティとの対話

これまでに、ＤＭＳ構成のコンポーネントおよび概念について述べた。ＤＭＳ（Ａ）またはＡ′は、プリンシパルが、ドキュメント空間でドキュメントと対話することが可能となるように設計されることに注意する必要がある。ブラウザ（たとえば、図１のブラウザ１２および図９のブラウザ９４）によって、プリンシパルが、属性との間、およびドキュメント属性によって編成される群との間において、文字通りの意味の会話をすることを可能とする、このような対話の一つの様式が提供される。

【０１６８】図１２は、使用中のブラウザの一例を示す。図には、四つの基本実体が示されている。ドキュメント１０２は、個別の実体として表示され、移動、除去および実行することができる。ドキュメント群は、包含するドキュメントを示す閉鎖された卵形１０４として、およびパイル（piles）１０６として、開かれる２つの形態で見える。パイルの概念は、Ｒ．Ｍａｎｄｅｒ、Ｇ．ＳａｌｏｍｏｎおよびＹ．Ｙ．Ｗｏｎｇによる論文"A Pile Metaphor for Supporting Casual Organization of Information"、PROC. ACM Conf. Human Factorsin Computing System, CHI 92（Monterrey, カリフォルニア州）１９９２年５月に記載されている。閉鎖型群をパイル１０６として表示するによって、閉鎖型群のサイズに関するキューを付与する本来の手段が提供される。流動群は、ユーザ動作と無関係に成長および縮小するので、この手段は特に有用である。

【０１６９】個別のプロパティをブラウザオブジェクトとして記憶することも可能であり、これは三角形１０８としてデスクトップ上に現れる。プロパティをブラウザと共に使用する際には、２つの規則がある。第一は、プロパティは、ドキュメントの上にドロップされ、これにより特定のプロパティをドキュメントに付加することができることである。第二は、個別プロパティをクエリー項目として使用できることである。

【０１７０】群は、従来のファイルシステムのフォルダおよびディレクトリと全く同様に、特定の他のドキュメント（サブ群を含む）を含むことができるのみでなく、群は、動的内容を特定するクエリー成分を含むこともできることを想起されたい。したがって、どの群の場合も、プリンシパルは、クエリー項目のセットを特定することができる。システム内のドキュメントは、クエリーと一致すると、群に含まれることになる（但し、明確に除外されたドキュメントを除く）。

【０１７１】クエリー項目は、従来のダイアログボックスを用いて特定することができるが、直接操作によっても、またプロパティアイコン１０８によっても特定できる（この実施形態では三角形であるが、他の図案を使用することもできる）。開放型群の上にプロパティをドラッグすることによって、その群のためのクエリー項目のリストに、そのプロパティを付加することができる。したがって、プロパティが"プロジェクト＝ＤＭＳ"である場合、その後、そのプロパティはドキュメントに付加することができるのみでなく、群上にドロップすることもできるので、"プロジェクト＝ＤＭＳ"は、その群に対する現行のクエリー項目のセットに付加される。

【０１７２】図１３に示すように、現行のクエリー項目のセットを表すプロパティアイコン１０８は、群オブジェクトの周上に出現する。これらのクエリー項目をドラッグして群から取り出すと、再度、これらのタームはクエリーから除去される。クエリー項目のこれらのアイコン表現は、ドラッグによりクエリーオブジェクトに付加およびそれから除去されるので、クエリーは別のスレッドで更新される。その結果、クエリーは、クエリー項目の操作に、即時に動的に応答し、ドキュメント空間上に構成可能なフィルタとして、文字通りのクエリーを付与する。

【０１７３】使用中のブラウザは、クエリーの生成ではなく群の操作と同様に、プリンシパルが実装するクエリー成分を有する群をさらに探索するように構成される。ブラウザの対話形式は、クエリーの作成および実行ではなく、ドキュメント空間の実装に基づく対話を記述する

ことを目的とする。前述した中で、クエリー項目によって操作の意味の付与が容易になることに注目したが、群の他の成分、包含および排他リストを用いて、操作の経験の支援を容易にできることは理解される。

【０１７４】前述した、流動群の包含および排除リストによって、支援を受けている対話型フィールドにとって重要である安定的な感受性が付与される。したがって、群の内容を動的に維持するクエリー成分に加えて、直接操作によって、包含および排他リストの使用が管理され、結果が修正される。ドキュメントをドラッグしてクエリー群から出すことによって、ドキュメントはその群の排他リストに付加される。このことは、このドキュメントは、クエリーが次回実行されるとき、群内に再び出現しないことを意味する（これは、バックグラウンドにおいて規則正しく起こる）。同様に、ドキュメントをドラッグして群に入れることは、それ以外の場合はドキュメントは群のコンポーネントとして包含されないので、ドキュメントを包含リストに加えるべきであることを意味する。この機構を用いて、プリンシパルは、プロパティをクエリーウィンドウにドロップしてプリンシパルが関心のある基本セットを表現するクエリーを作成し、次に、特定の項を付加または除去することによって結果をより精密化することができる。結果として得られる群は、動的に実行されたクエリーより「実際」の実体により近く感じられるが、これらのドキュメントがシステムに付加されるとき、重要な新ドキュメントは、まだ包含されている。

【０１７５】複数のドキュメントを作業領域に出現させるが、二つのドキュメントが同じ出現場所に出現する状況を避けることは可能である。すなわち、一つのドキュメントは、所定の群に一回以上、またはデスクトップに一回以上、出現できない。ユーザが、ドキュメントが既に出現した出現場所に移動させようとする場合、ユーザがマウスを放すと「第二」外観が第一外観と合体する。

【０１７６】前述したように、また図１４に示すように、ディスプレイスクリーン１１２のダイアログボックス１１０を用いて、ドキュメント上のプロパティを変更することもできる。特に、図１４に示すように、群"goodies"は、ドキュメント"generation-of-bits-html"を含む。この情報は、ディスプレイ１１４に示される。ドキュメント"generation-of-bits-html"が強調表示されると、プロパティリストウィンドウディスプレイ１１６は、そのドキュメントに付加されている静的プロパティ１１８および動的プロパティ１２０を表示する。プロパティは、ダイアログボックス１１０の使用によって、付加または除去あるいは他の場合は探索することができる。プロパティが任意のデータを含むことができることを、表示されるプロパティリストが示していることも注目される。プロパティリストウィンドウディスプレイ１１６は、サブプロパティまたは子プロパティ１２４、１２６が付加されている親プロパティ１２２を含む。この図にはさらに、複数のプロパティが同じ名を持つことができることを示す（１２８、１３０）。

【０１７７】以上、本発明を好適な実施形態を参照して述べた。この明細書を読み理解するとき、他者が変形および異形を思いつくことは明白である。すべてのこのような変形および異形は、特許請求の範囲内またはその同等物の範囲内に由来する限りにおいて、本発明に含まれるものとする。

【０１７８】

【発明の効果】以上説明したように、本発明によれば、ドキュメントへのプロパティの付加により、ドキュメント内容の記憶位置をドキュメントの管理から分離することができるので、ユーザは、ドキュメントの編成、記憶および検索において柔軟性を持つことができ、ドキュメントの管理を、容易、迅速かつ正確に行うことができる。

【図面の簡単な説明】

【図１】 本発明のプロパティの構想と比較した階層保管メカニズムを示す説明図である。

【図２】 ユーザとオペレーティングシステムとの間の通信チャネル内に挿入された本発明にかかるドキュメント管理システムのブロック図である。

【図３】 コンピュータシステムで実行される本発明にかかるドキュメント管理システムの説明図である。

【図４】 従来のファイルシステムの概念を示す説明図である。

【図５】 ドキュメントの内容がドキュメントのプロパティから分離されている本発明の概念を示す説明図である。

【図６】 ドキュメントの内容がドキュメントのプロパティから分離されている本発明の概念を示す説明図である。

【図７】 本発明におけるプロパティの通知態様の概念を示す説明図である。

【図８】 本発明におけるコンピュータシステム内の操作とプロパティとの関係の概要を示す説明図である。

【図９】 本発明の一実施形態にかかるドキュメント管理システムの概略を示す説明図である。

【図１０】 本発明におけるドキュメントへのプロパティ付加の概念を示す説明図である。

【図１１】 本発明のプロパティに関するコードの一例を示す図である。

【図１２】 本発明におけるコンピュータスクリーン上に示されたブラウザの一例を示す図である。

【図１３】 図１２の一部拡大図である。

【図１４】 本発明におけるドキュメント群のリストおよび本発明の一つに付加されるプロパティのリストの一例を示す図である。

【符号の説明】

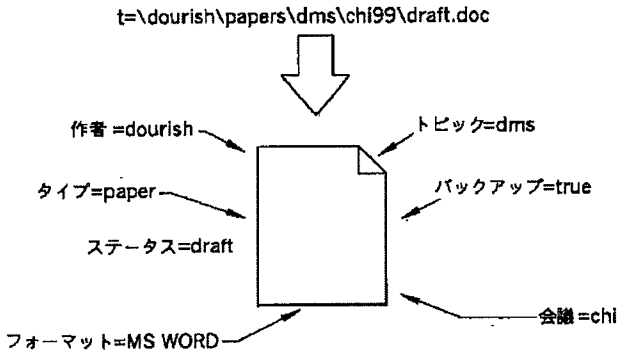
35

１０ａ～１０ｎ　ＤＭＳ認識アプリケーション、１２，９４，９６ａ　ブラウザ、１４ａ～１４ｎ，６０ｃ　保管場所、１６，８８ａ～８８ｎ　ビットプロバイダ、１８ａ～１８ｎ，６０ａ　カーネル、２０ａ～２０ｎ，６０ｂ，６２ａ～６２ｎ，１０２　ドキュメント、２２，１１８　静的プロパティ、２４，６４ａ～６４ｎ，９６ｃ，１２０　動的プロパティ、２６　基本プロパティ、２８，３０，３２　リンク、４０　識別情報、４２，４６　内容、４４ａ～４４ｎ，６６ａ～６６ｎ　外部送信

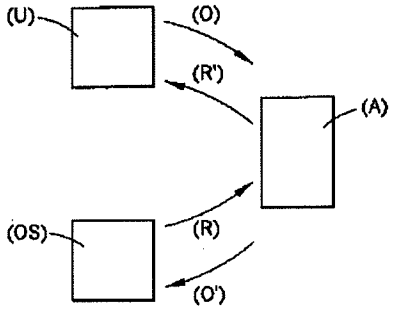
36

元、７０ａ～７０ｎ　通知、７６ａ～７６ｄ　操作、８０ドキュメント層、８２ａ，８２ｂ　インターフェース、８２ｃ　トランスレータ、８３　Ｊａｖａビーンズ、８４，９６ｂ　アプリケーション、８６　サービスコンポーネント（プロパティ発生器）、９２　プロパティデータベース、９６ｄ，９６ｅ，９６ｆ　プロパティ発生器、１０４　ドキュメント群、１０６　パイル、１０８　個別プロパティ（アイコン）。

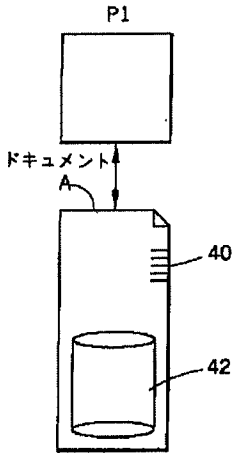
【図１】



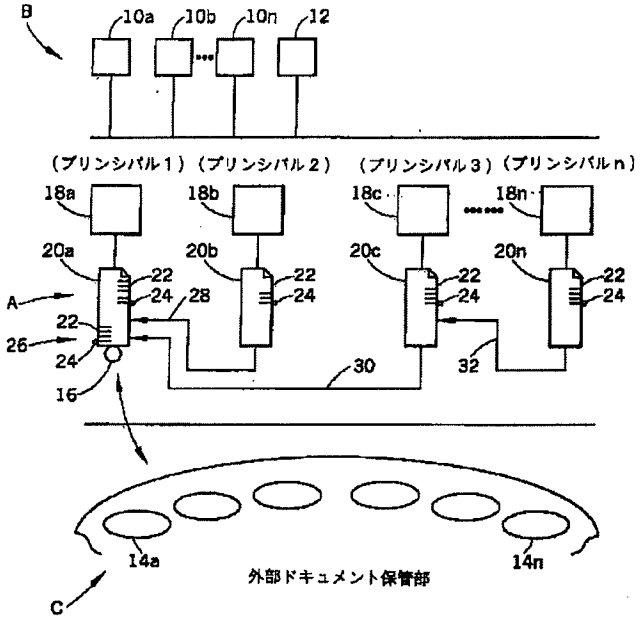
【図２】



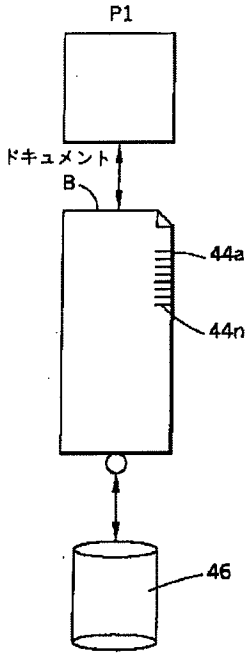
【図４】



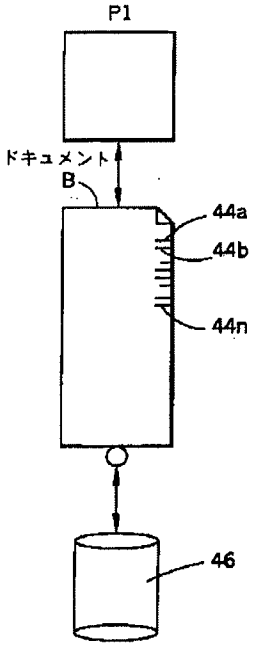
【図３】



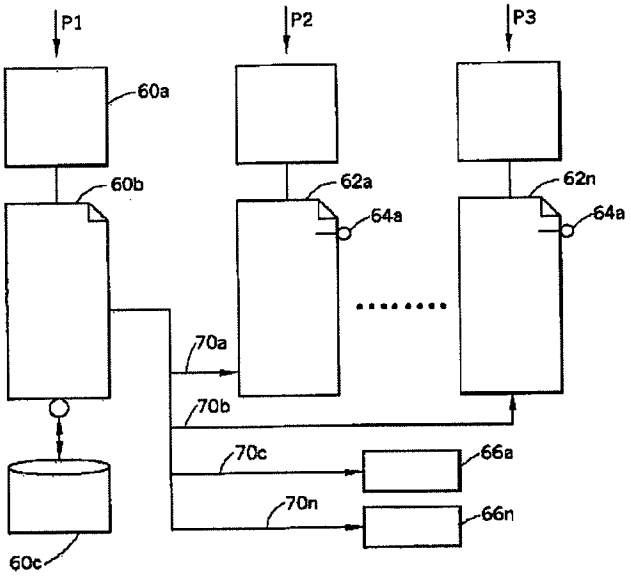
【図５】



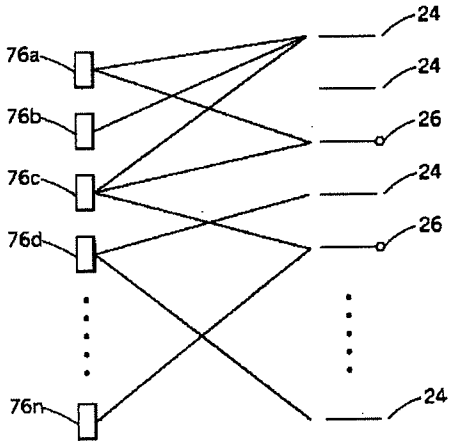
【図６】



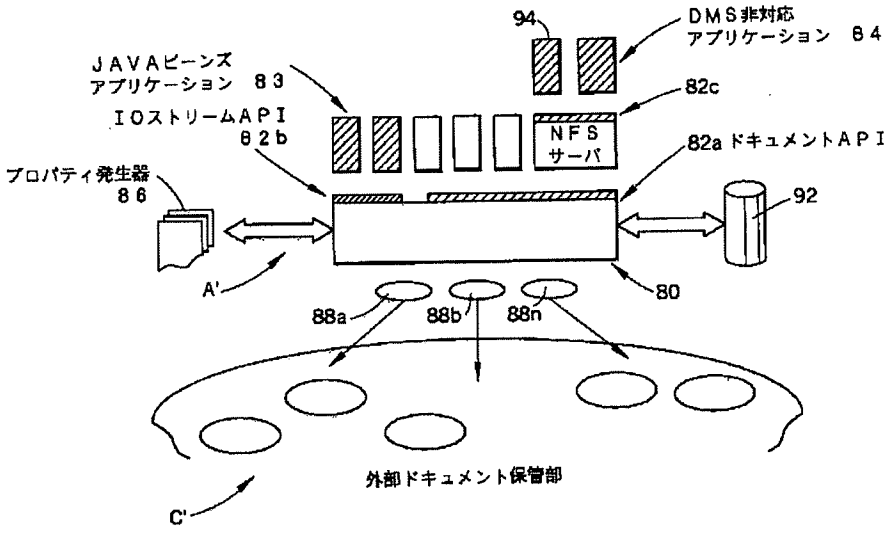
【図７】


P1
P2
P3
60a
60b
62a
62n
64a
64a
70a
70b
70c
70n
66a
66n
60c

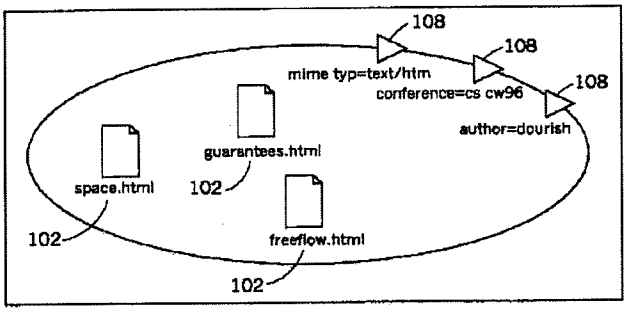
【図８】


76a
76b
76c
76d
76n
24
24
26
24
26
24

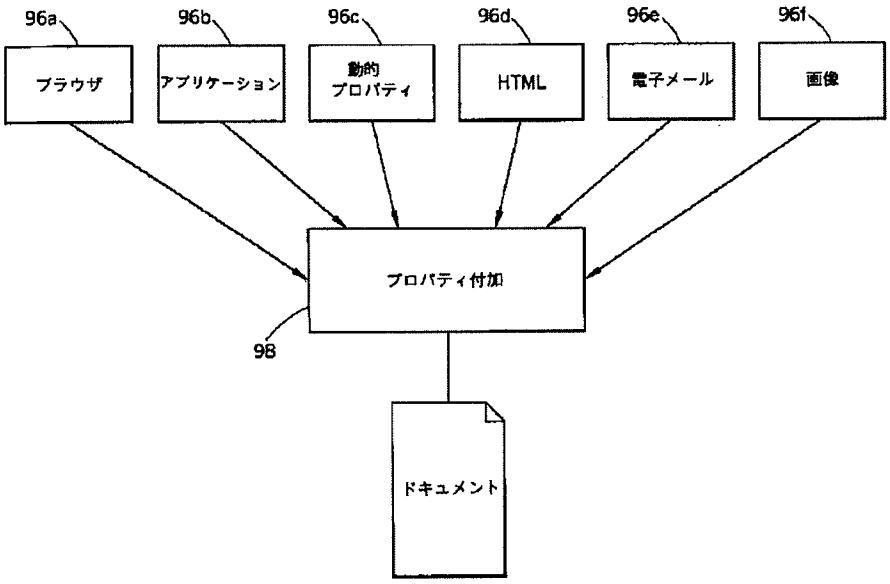
【図９】


94
DMS非対応
アプリケーション 84
JAVAビーンズ
アプリケーション 83
IOストリームAPI
82b
82c
NFS
サーバ
82a ドキュメントAPI
プロパティ発生器
86
92
A'
80
88a
88b
88n
外部ドキュメント保管部
C'

【図１３】


108
108
108
mime typ=text/htm
conference=cs cw96
author=dourish
guarantees.html
space.html
102
102
freeflow.html
102

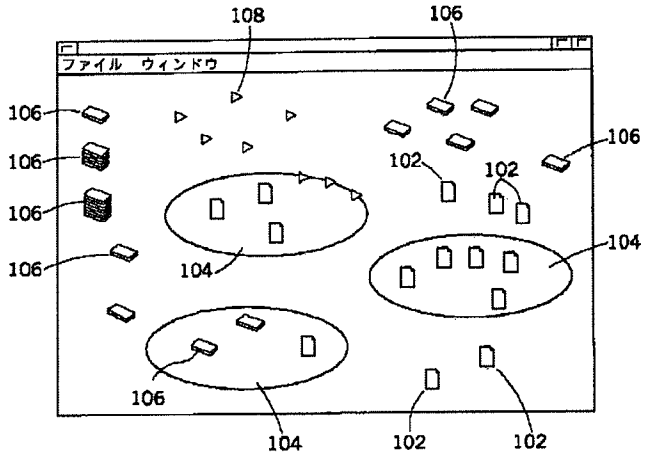
【図１０】



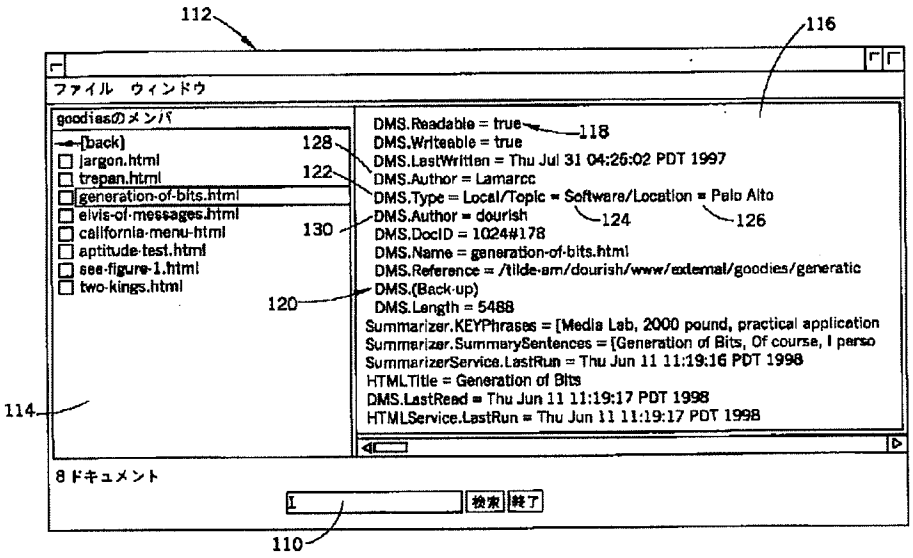
【図１１】

```
パッケージ試験;

import java.util.Enumeration;
import DMS.db.*;
import DMS.db.client.*;

public class testprogram {
public static void main (String args []) {

   // ＤＭＳデータベースへの接続
 DMS.startMysqlDatabase ();

 //要求するドキュメントに対するクエリー作成、
 //データベース中の一致する全ドキュメントを含むＤｏｃＬｉｓｔを作成
 Query q = new Query("project = DMS");
 DocList docs = q.find();


 //ｄｏｃＬｉｓｔの反復、各ドキュメントに対するプロパティ設定
 for (Enumeration e = docs. elements(); e.hasMoreElements ();) {
 DMSItem p = (DMS.Item e.nextElement();
 p.setAttribute("coolness", "high");
 }
 //新しい群の作成
 Collection c = null;
 try {
  c = DMS.createCollection ();
 } catch (DMS.DBException e) {
  System.err.println (e) ;
  System.exit(1);
 }

 //群の名設定
 c.setAttribute (DMS Item.NameField, "cool documents");

 //群の帰属関係を特定するクエリーの設定
 c.setQuery(new Query ("coolness = high");
 System.out.println(c.getDocuments().size() + = items");

 System.exit(0);

 }
}
```

【図１２】



【図１４】



フロントページの続き

(51) Int. Cl.[7]	識別記号	F I	ﾃｰﾏｺｰﾄﾞ(参考)
		G06F 15/403	360Z

(72)発明者　ジェームズ　ピー　ドーリッシュ
アメリカ合衆国　カリフォルニア州　サン　フランシスコ　ドロレス　ストリート　1040　＃105
(72)発明者　ワレン　ケイ　エドワーズ
アメリカ合衆国　カリフォルニア州　サン　フランシスコ　グエレロ　ストリート　1527
(72)発明者　アンソニー　ジー　ラマルカ
アメリカ合衆国　カリフォルニア州　レッドウッド　シティ　コネティカット　ディアール　1727
(72)発明者　ジョン　オー　ランピング
アメリカ合衆国　カリフォルニア州　ロス　アルトス　エバ　アベニュー　1299
(72)発明者　マイケル　ピー　サリスブリー
アメリカ合衆国　カリフォルニア州　マウンテン　ビュー　シャワーズ　ドライブ　イー254　49
(72)発明者　ダグラス　ビー　テリー
アメリカ合衆国　カリフォルニア州　サン　カルロス　タスカー　レーン　23
(72)発明者　ジェームズ　ディー　ソーントン
アメリカ合衆国　カリフォルニア州　レッドウッド　シティー　ベイ　ロード　3312